软件质量保证
与软件测试技术

软件测试概述

万立中
www.wanlizhong.com

软件测试概述

1.1 软件测试背景

1.2 软件测试基础理论

1.3 软件开发过程

1.4 软件开发与软件测试的关系

1.5 软件测试过程

1.6 软件质量保证概要

目标

- 正确理解软件测试的背景、软件缺陷和故障的概念
- 正确理解软件测试的意义
- 正确理解软件开发过程与软件测试的关系
- 正确理解软件质量的概念及质量保证体系

1.1 软件测试背景

1.1.1 软件可靠性问题

1.1.2 软件缺陷与故障

1.1.3 软件缺陷产生的原因

Return

1.1.1 软件可靠性问题

- 因软件设计故障与因计算机硬件设计故障而引发的系统失效的比例大约是：10:1
- 运行软件的驻留故障密度（每千行代码的故障数目）：
  ——要求很高的关键财务或财产软件为：每千行代码1～10个故障
  ——关键的生命软件为：每千行代码0.01～1个故障
- IEEE将软件可靠性定义为：系统在特定环境下，在给定的时间内无故障运行的概率。
- 软件可靠性是对软件在设计、开发以及所预定的环境下具有能力的置信度的一个度量，是衡量软件质量的主要参数之一。而软件测试则是保证软件质量、提高软件可靠性的最重要手段。

1.1.2 软件缺陷与故障

1、软件缺陷和软件故障案例

- 案例1　美国迪斯尼公司的狮子王游戏软件bug
  兼容性问题
- 案例2　美国航天局火星登陆事故
  系统测试　衔接问题
- 案例3　跨世纪“千年虫”问题
- 案例4　爱国者导弹防御系统炸死自家人
  系统时钟误差积累
- 案例5　Windows 2000 中文输入法漏洞

上述所有实例中的软件问题在软件工程或软件测试中都被称为软件缺陷或软件故障。

软件缺陷与故障（续）

2、软件缺陷的定义

（1）软件未达到产品说明书中已经标明的功能；

（2）软件出现了产品说明书中指明不会出现的错误；

（3）软件未达到产品说明书中虽未指出但应当达到的目标；

（4）软件功能超出了产品说明书中指明的范围；

（5）软件测试人员认为软件难以理解、不易使用，或者最终用户认为该软件使用效果不良。

- 举例：计算器内的嵌入式软件

软件缺陷与故障（续）

3、软件缺陷的特征

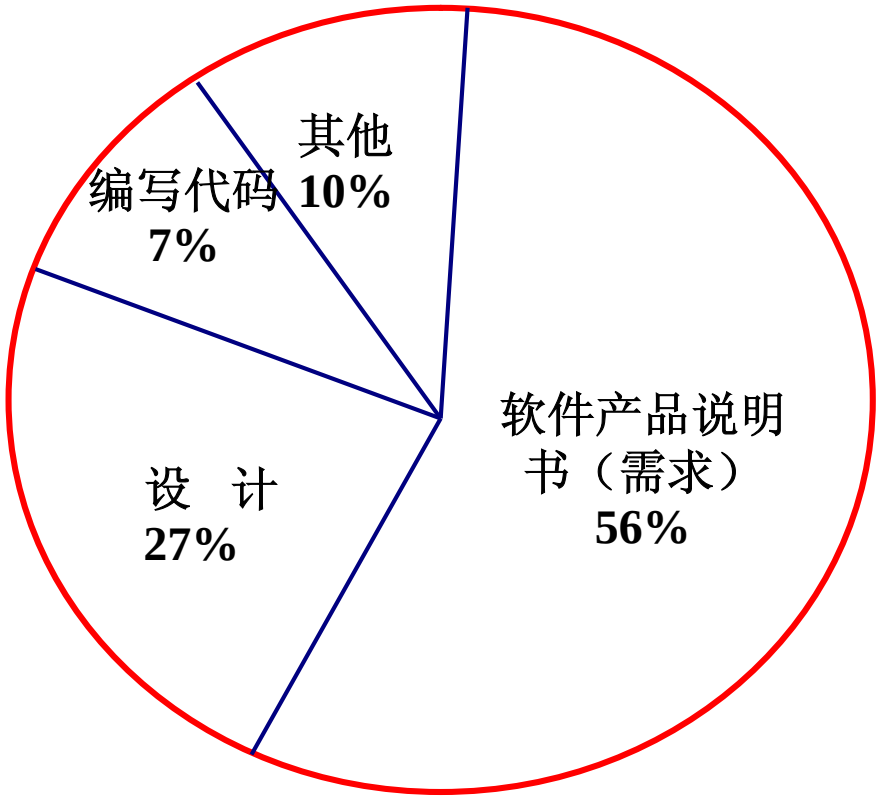
- “看不到”

  ——软件的特殊性决定了缺陷不易看到

- “看到但是抓不到”

  ——发现了缺陷，但不易找到问题发生的原因所在

1.1.3 软件缺陷产生的原因



Return

1-1 软件缺陷产生的原因分布

1.2 软件测试基础理论

1.2.1 软件测试的定义

1.2.2 软件测试的基本理论

1.2.3 软件测试和缺陷修复的代价

1.2.4 软件测试技术概要

Return

1.2.1 软件测试的定义

1、软件测试的定义

软件测试就是在软件投入运行前，对软件需求分析、设计规格说明和编码实现的最终审查，它是软件质量保证的关键步骤。通常对软件测试的定义有两种描述：

- **定义1：** 软件测试是为了发现错误而执行程序的过程。
- **定义2：** 软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计的一批测试用例，并利用这些测试用例运行程序以及发现错误的过程，即执行测试步骤。

软件测试的定义（续）

- **测试：**所谓测试的含义，首先是一项活动，在这项活动中某个系统或组成的部分将在特定的条件下运行，结果将被观察和记录，并对系统或组成部分进行评价。测试活动有两种结果：找出缺陷和故障，或显示软件执行正确。测试是一个或多个测试用例的集合。
- **测试用例：**所谓测试用例是为特定的目的而设计的一组测试输入、执行条件和预期的结果；测试用例是执行测试的最小实体。
- **测试步骤：**测试步骤详细规定了如何设置、执行、评估特定的测试用例。

软件测试的定义（续）

2、软件测试的基本问题

- 软件生命周期：一个软件生命周期包括制定计划、需求分析定义、软件设计、程序编码、软件测试、软件运行、软件维护、软件停用等8个阶段。
- 软件测试的对象：

——软件测试不等于程序测试。

——软件测试贯串于软件定义和开发的整个过程。

——软件开发过程中所产生的需求规格说明、概要设计规格说明、详细设计规格说明以及源程序都是软件测试的对象。

软件测试的定义（续）

2、软件测试的基本问题（续）

- 软件测试在软件生命周期中横跨两个阶段：

  第一个阶段：单元测试阶段，即在每个模块编写出以后所做的必要测试。

  第二个阶段：综合测试阶段，即在完成单元测试后进行的测试，如集成测试、系统测试、验收测试。

- 软件测试涉及的关键问题包括四个方面：

  （1）测试由谁来执行。　　（2）测试什么。

  （3）什么时候进行测试。　（4）怎样进行测试。

1.2.2 软件测试的基本理论

1、软件测试的目的

（1）测试是程序的执行过程，目的在于发现错误；不能证明程序的正确性，除非仅处理有限种情况。

（2）检查系统是否满足需求也是测试的期望目标。

（3）一个好的测试用例在于发现了还未曾发现的错误；一次成功的测试则是发现了错误的测试。

注意：测试无法说明错误不存在，只能说明软件错误已出现。

软件测试的基本理论（续）

2、软件测试的原则

（1）尽早地和及时地测试；

（2）测试用例应当由测试数据和与之对应的预期结果这两部分组成；

（3）在程序提交测试后，应当由专门的测试人员进行测试；

（4）测试用例应包括合理的输入条件和不合理的输入条件；

（5）严格执行测试计划，排除测试的随意性；

（6）充分注意测试当中的群体现象；

（7）应对每一个测试结果做全面的检查；

（8）保存测试计划、测试用例、出错统计和最终分析报告，为维护工作提供充分的资料。

软件测试的基本理论（续）

3、软件测试的分类

软件测试按照不同的划分方法，有不同的分类：

- 按照软件测试用例的设计方法而论，软件测试可以分为白盒测试法和黑盒测试法。
- 按照软件测试的策略和过程来分类，软件测试可分为单元测试、集成测试、系统测试、验证测试和确认测试。

软件测试的基本理论（续）

4、测试信息流程

测试信息流程如图1-2所示。测试过程中需要三类输入：软件配置、测试配置和测试工具。

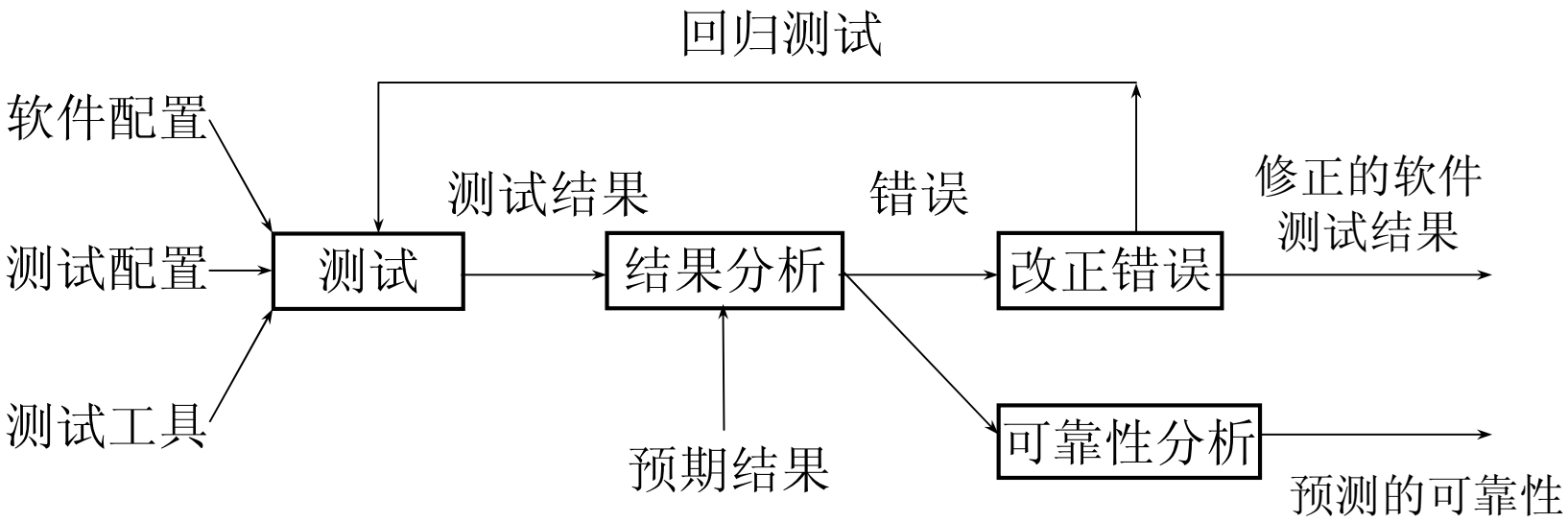


图1-2 测试信息流程

软件测试的基本理论（续）

5、软件测试的周期性

软件测试的周期性是“测试->改错->再测试->再改错”这样一个循环过程，如下图1-3所示。

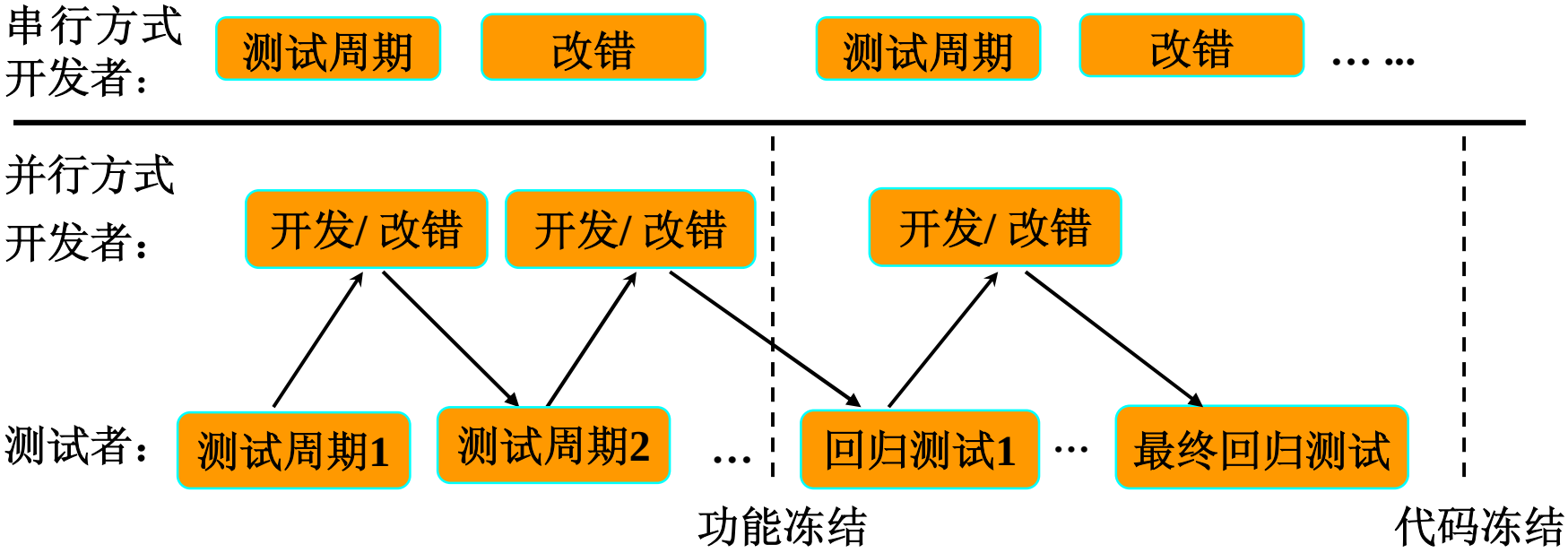


图1-3 软件测试的周期性

软件测试的基本理论（续）

6、测试停止的依据（标准）

- 第一类标准：测试超过了预定时间，则停止测试。
- 第二类标准：执行了所有的测试用例，但并没有发现故障，则停止测试。
- 第三类标准：使用特定的测试用例设计方案作为判断测试停止的基础。
- 第四类标准：正面指出停止测试的具体要求，即停止测试的标准可定义为查出某一预订数目的故障。
- 第五类标准：根据单位时间内查出故障的数量决定是否停止测试。

1.2.3 软件测试和缺陷修复的代价

- 软件在从需求、设计、编码、测试一直到交付用户公开使用后的过程中，都有可能产生和发现缺陷。随着整个开发过程的时间推移，更正缺陷或修复问题的费用呈几何级数增长。

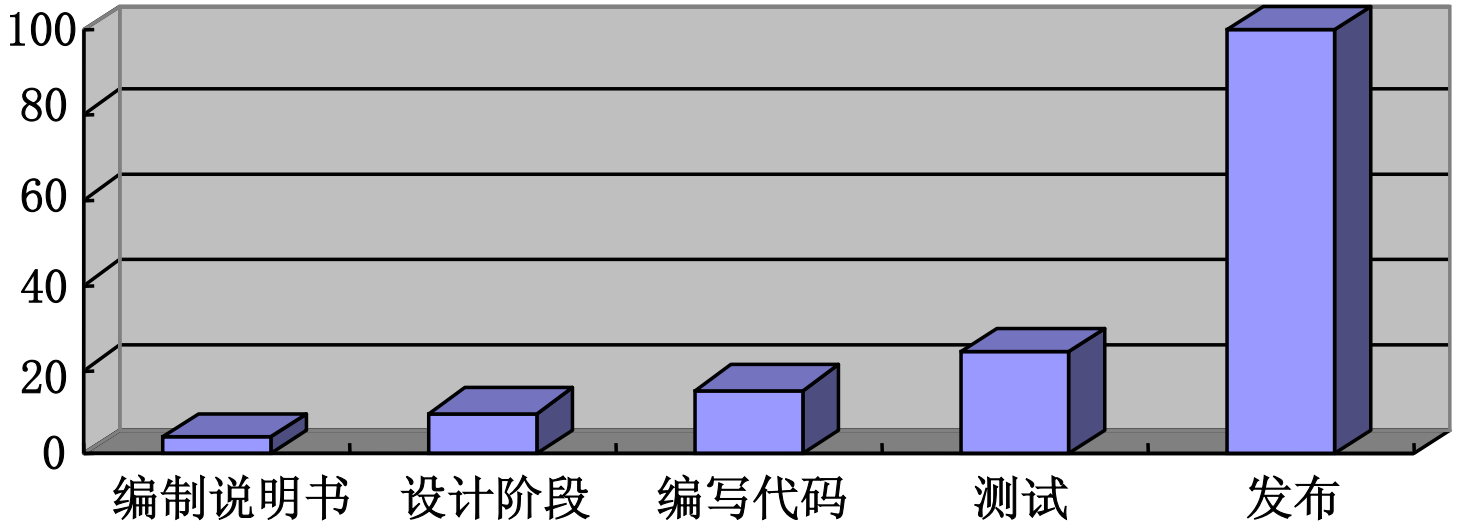


图1-4 软件缺陷在不同阶段发现时修复的费用示意图

1.2.4 软件测试技术概要

- 软件测试的策略：就是测试将按照什么样的思路和方式进行。通常，软件测试要经过单元测试、集成测试、确认测试、系统测试以及验收测试。
- 软件测试技术：

（1）白盒测试和黑盒测试

（2）静态测试和动态测试

（3）传统测试方法和面向对象测试的方法

（4）特定环境及应用的测试

测试技术

不实际运行程序，而是通过检查和阅读等手段来发现错误并评估代码质量的软件测试技术。也称为静态分析技术。

实际运行程序，并通过观察程序运行的实际结果来发现错误的软件测试技术。

在不知道程序内部结构，只知道程序规格的情况下采用的测试技术或策略。

在知道程序内部结构的情况下采用的测试技术或策略。

软件测试技术

静态测试

动态测试

黑盒测试

白盒测试

在测试过程中，选择足够的测试用例，使得每一个可执行语句至少被执行一次。

在测试过程中，选择足够的测试用例，使得程序中的每一个分支判断的每一种可能结果都至少被执行一次。

开发组内部进行的，采用讲解、提问并使用Checklist方式进行的查找错误的活动。

性能

在测试过程中，选择足够的测试用例，使得程序中的每一条可能执行的路径都至少执行一次。

软件测试技术概要（续）

- 软件测试技术的发展趋势：

（1）软件验证技术

（2）静态测试分析技术

（3）测试数据的选择——主要对测试用例进行选择

通常从下面几个方面评价测试用例的质量：

检测软件缺陷的有效性、测试用例的可重用性、测试用例的经济性、测试用例的可维护性

（4）集成化测试——研究如何实现软件测试的自动化过程以及相关的一系列内容。

Return

1.3 软件开发过程

1.3.1 软件产品的组成

1.3.2 软件开发项目组

1.3.3 软件开发模式

Return

1.3.1 软件产品的组成

1、软件产品需要各种开发投入

产品说明书、产品审查、设计文档、进度计划、上一版本信息反馈、商业竞争对手的同类软件产品情况、客户调查、易用性数据、观察与感受说明书

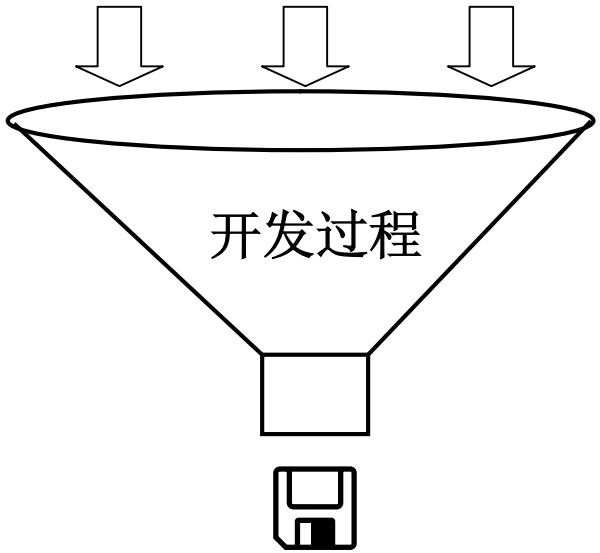


图1-5 获得软件产品的工作示意图

软件产品的组成（续）

2、客户需求

客户需求包括对客户调查所收集的详细信息、以前软件的使用情况及存在的问题、竞争对手的软件产品信息等等。通过分析客户需求，可以确定将要开发的软件产品应该具有哪些功能。

3、产品说明

产品说明书的作用就是对客户需求信息进行综合描述，并包括用户没有提出、但软件产品本身必须要实现的要求，从而针对产品进行定义并确定其功能。

软件产品的组成（续）

4、设计文档

- 构架。即产生描述软件整体设计的文档，包括软件所有主要部分的描述以及相互间的交互方式。
- 数据流示意图。表示数据在程序中如何流动的正规示意图。通常由圆圈和线条组成，所以也称为泡泡图。
- 状态变化示意图。将软件分解为基本状态或者条件的另一种正规示意图，表示不同状态之间的变化的方式。
- 流程图。用图形描述程序逻辑的最常用方式之一。根据详细的流程图编写程序代码简单方便。
- 注释代码。代码注释是便于维护代码的程序员掌握代码的内容和执行方式。

软件产品的组成（续）

5、测试文档

一般测试文档所包含的内容：

- 测试计划。描述用于验证软件是否符合产品说明书和客户需求的整体方案。
- 测试用例。依据测试的项目，并描述验证软件的详细步骤。
- 软件测试报告。描述依据测试用例找出的问题，通常提交测试报告。
- 归纳、统计和总结。采用图表、表格和报告等形式来描述整个测试过程。

软件产品的组成（续）

6、开发进度表

软件项目的开发进度通常使用Gantt图表来进行描述。

7、软件产品组成部分

（1）程序代码　（2）帮助文件　（3）用户手册

（4）样本和示例　（5）标签　（6）产品支持信息

（7）图表和标志　（8）错误信息　（9）广告与宣传材料

（10）软件的安装　（11）软件说明文件

（12）测试错误提示信息

1.3.2 软件开发项目组

- 项目管理经理：全程负责整个软件项目的开发。
- 系统设计师：设计整个系统构架或软件构思。
- 程序员：负责设计、编写程序，并修改软件中的缺陷。
- 软件测试员/测试师：负责找出并报告软件产品的问题，与开发组密切合作，进行测试并报告发现的问题。
- 技术制作、用户助手、用户培训员、手册编写和文件档案专员：负责编写软件产品附带的文件和联机文档 。
- 结构管理和制作人员：负责将程序员编写的全部文档资料合并成一个软件包 。

1.3.3 软件开发模式

1、大棒开发法

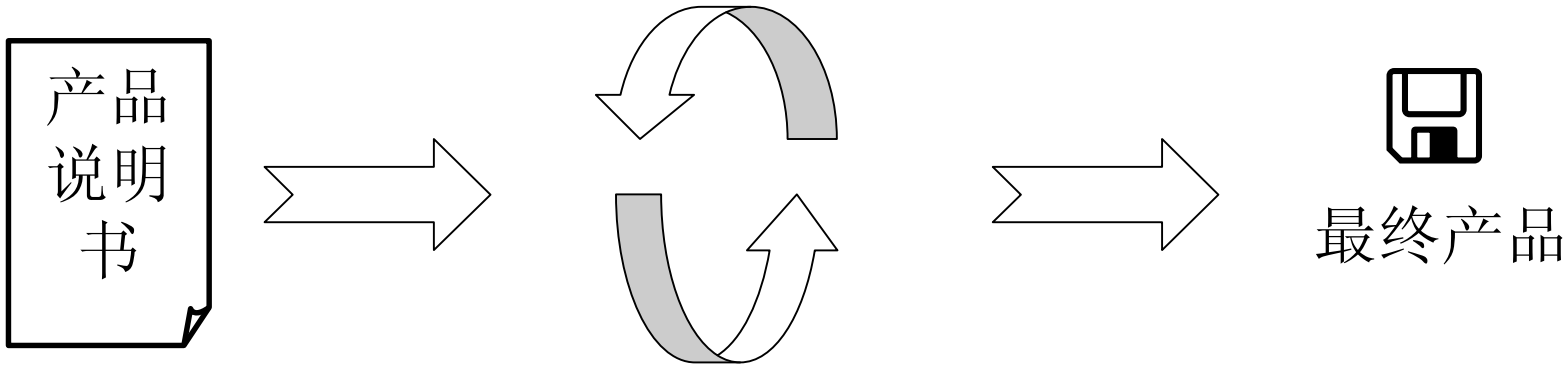
- 源于能量爆发创造宇宙，万物都由能量和物质积聚而成的理论，但如果不是遵循某种正确的排列和组合，形成的将不是预先期望的事物。大棒模式与上述理论一样：一大堆能量（这里指开发软件所需的人力和物力）放在一起，巨大的能量进行释放，通常的结果可能是产生了优秀的软件产品或成为一堆“废品”（不成功的软件）。
- 优点：思路简单， 通常可能是开发者的“突发奇想”
- 缺点：开发过程是非工程化的，随意性大
- 关于测试：有的较简单，有的则非常困难

软件开发模式（续）

2、边写边改法

- 采用边写边改法的软件开发通常只是有了比较粗略的想法就开始进行简单的设计、然后进行较长的反复编写、测试与修复这样一个循环的过程。在认为无法更精细的描述软件产品要求时，就发布产品。
- 优点：能够较为迅速的展现成果，适合需要快速制作而且用完就扔的小项目，如示范程序、演示程序等。
- 缺点：其编码和测试可能将是长期的循环往复的过程。

软件开发模式（续）



代码编制、测试、修复

图1-6 边写边改开发模式

软件开发模式（续）

3、瀑布法

- 瀑布模式是将软件生命周期的各项活动，规定为按照固定顺序相连的若干个阶段性工作，形如瀑布流水，最终得到软件产品。
- 优点：易于理解；调研开发的阶段性；强调早期计划及需求调查；确定何时能够交付产品及何时进行评审与测试。
- 缺点：需求调查分析只进行一次，不能适应需求变化；顺序的开发流程，使得开发中的经验教训不能反馈到该项目的开发中去；不能反映出软件开发过程的反复与迭代性；没有包含任何类型的风险评估；开发中出现的问题直到开发后期才能够显露，因此失去及早纠正的机会。

软件开发模式（续）

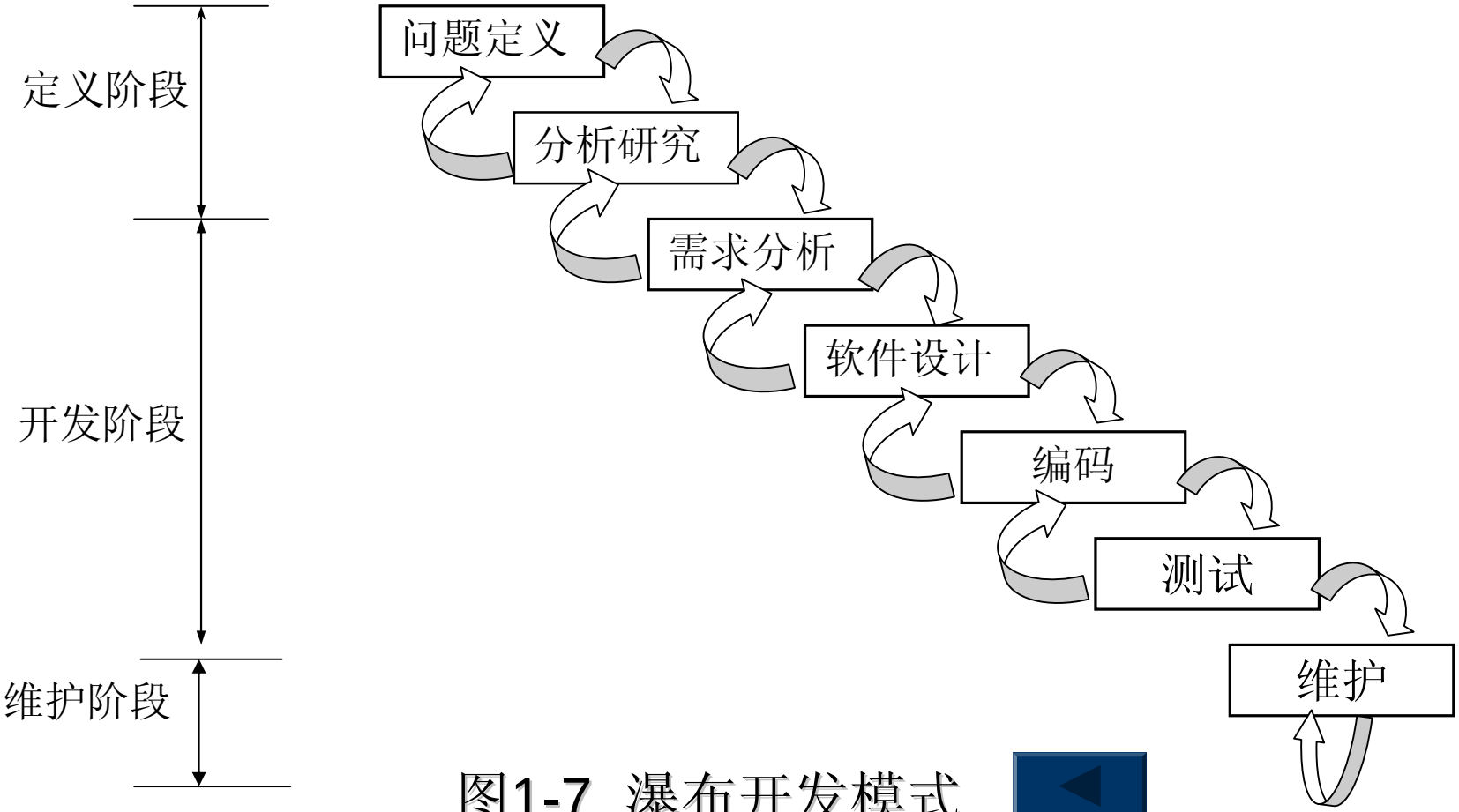


图1-7 瀑布开发模式

软件开发模式（续）

4、快速原型法

根据客户需求在较短的时间内解决用户最迫切解决的问题，完成可演示的产品。这个产品只实现最重要功能，在得到用户的更加明确的需求之后，原型将丢弃。

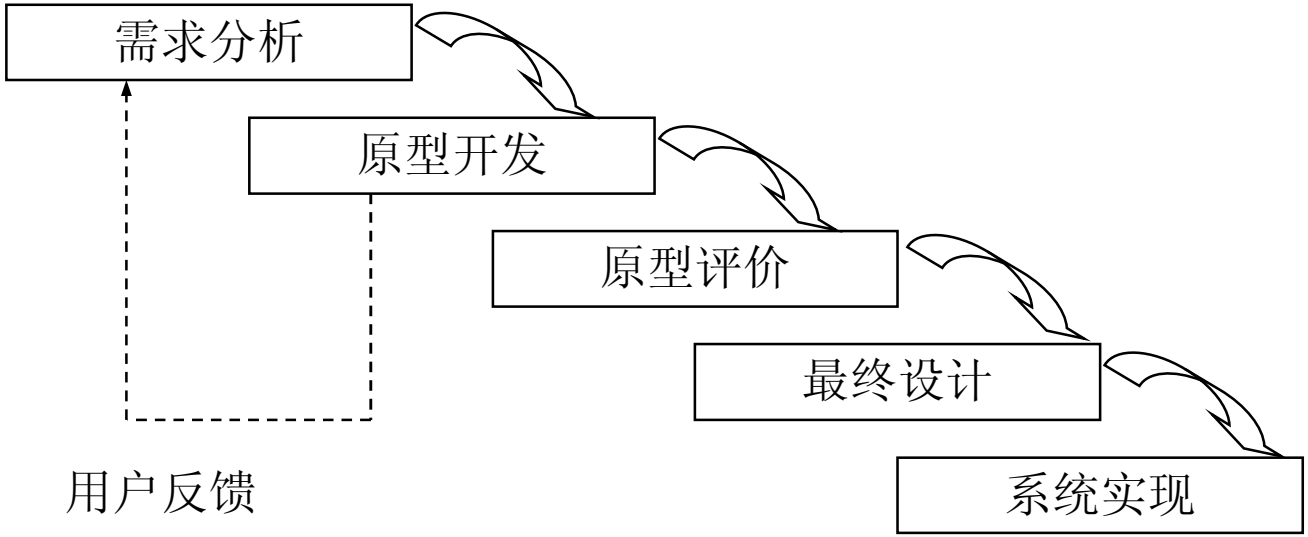


图1-8 快速原型开发模式

软件开发模式（续）

5、螺旋模式法

- 螺旋模式是瀑布模式与边写边改演化模式相结合，并加入风险评估所建立的软件开发模式。
- 主要思想是在开始时不必详细定义所有细节，而是从小开始，定义重要功能，尽量实现，接受客户反馈，进入下一阶段，并重复上述过程，直到获得最终产品。
- 每一螺旋（开发阶段）包括5个步骤：①确定目标，选择方案和限制条件。 ②对方案风险进行评估，并能解决风险。 ③进行本阶段的开发和测试。 ④计划下一阶段。 ⑤确定进入下阶段的方法。
- 优点：严格的全过程风险管理；强调各开发阶段的质量；提供机会评估项目是否有价值继续下去。

软件开发模式（续）

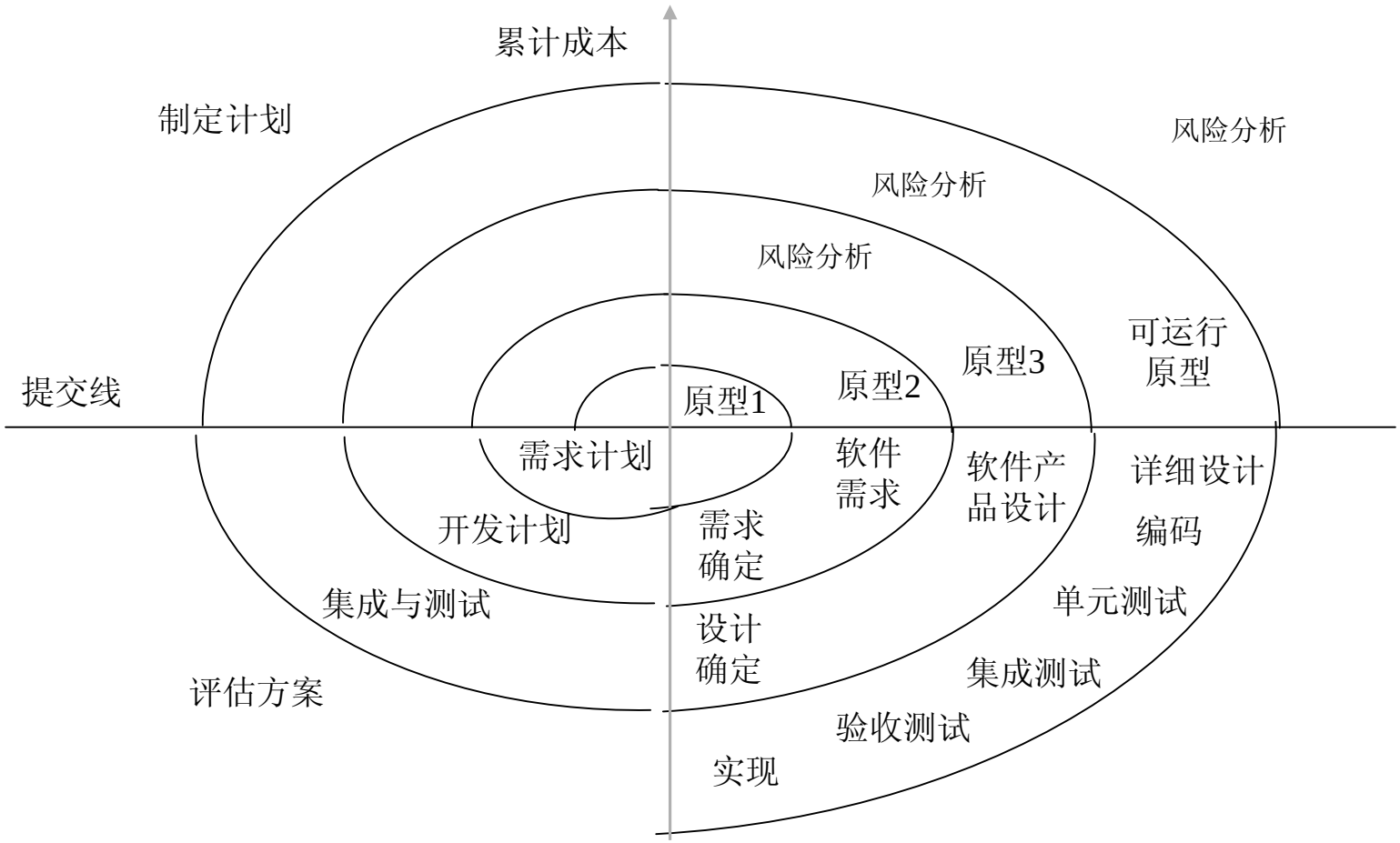


图1-9 螺旋开发模式

1.4 软件开发与软件测试的关系

1、测试与开发各阶段的关系

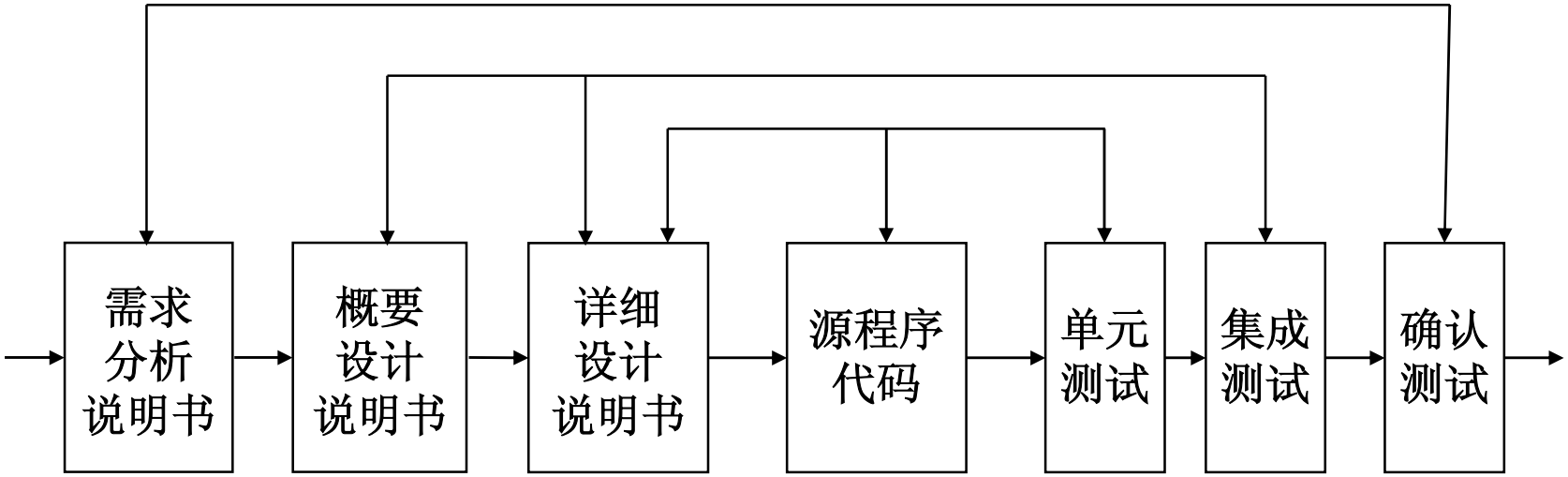


图1-10　软件测试与软件开发过程的关系

Return

软件开发与软件测试的关系（续）

测试在开发阶段的作用：

- 项目规划阶段：负责从单元测试到系统测试的整个测试阶段的监控。
- 需求分析阶段：确定测试需求分析、系统测试计划的制定，评审后成为管理项目。
- 详细设计和概要设计阶段：确保集成测试计划和单元测试计划完成。
- 编码阶段：由开发人员进行自己负责部分的测试代码。在项目较大时，由专人进行编码阶段的测试任务。
- 测试阶段（单元、集成、系统测试）：依据测试代码进行测试，并提交相应的测试状态报告和测试结束报告。

软件开发与软件测试的关系（续）

2、测试与开发的并行性

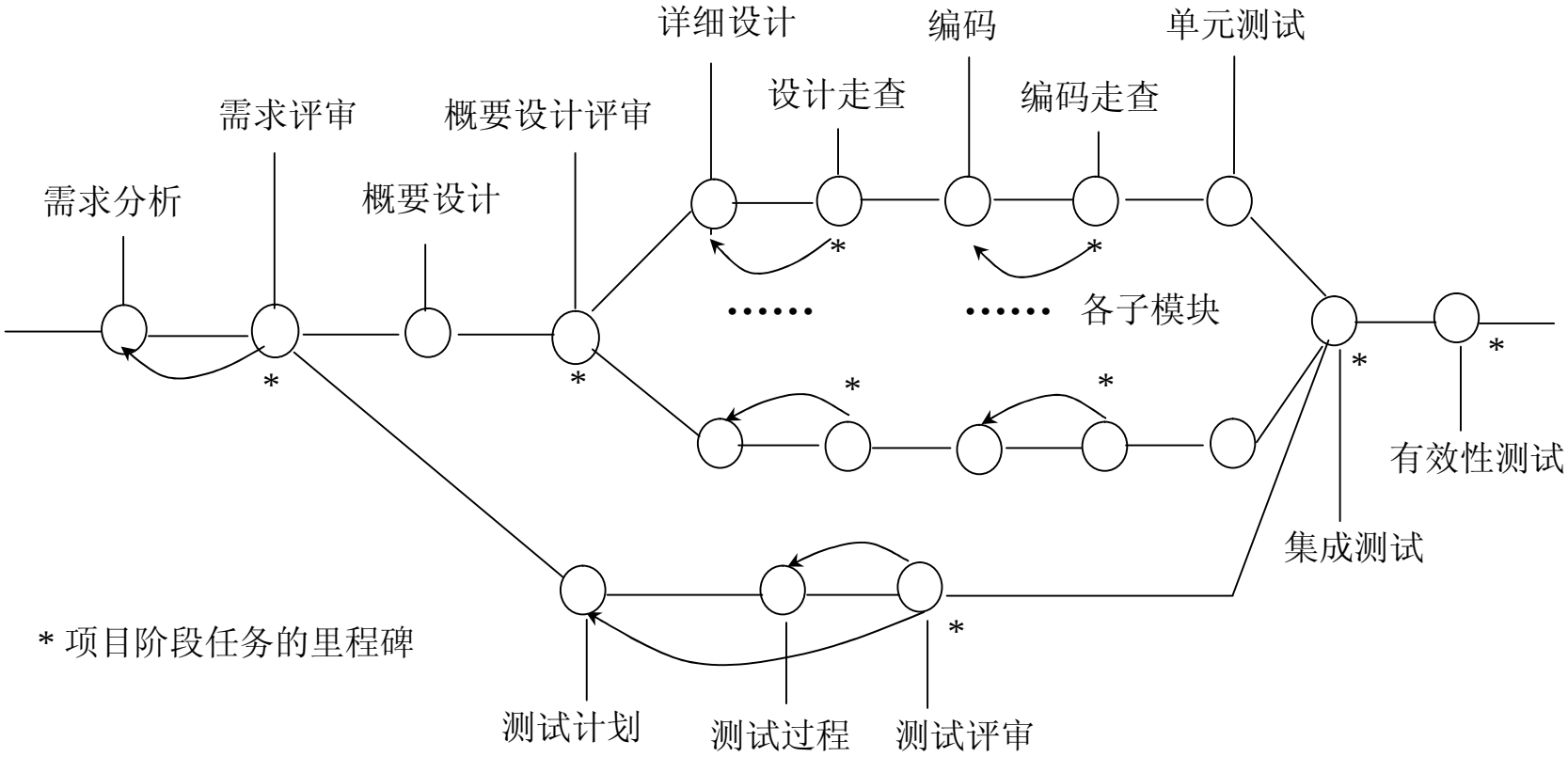


图1-11 软件测试与软件开发的并行性

软件开发与软件测试的关系（续）

3、完整的软件开发流程

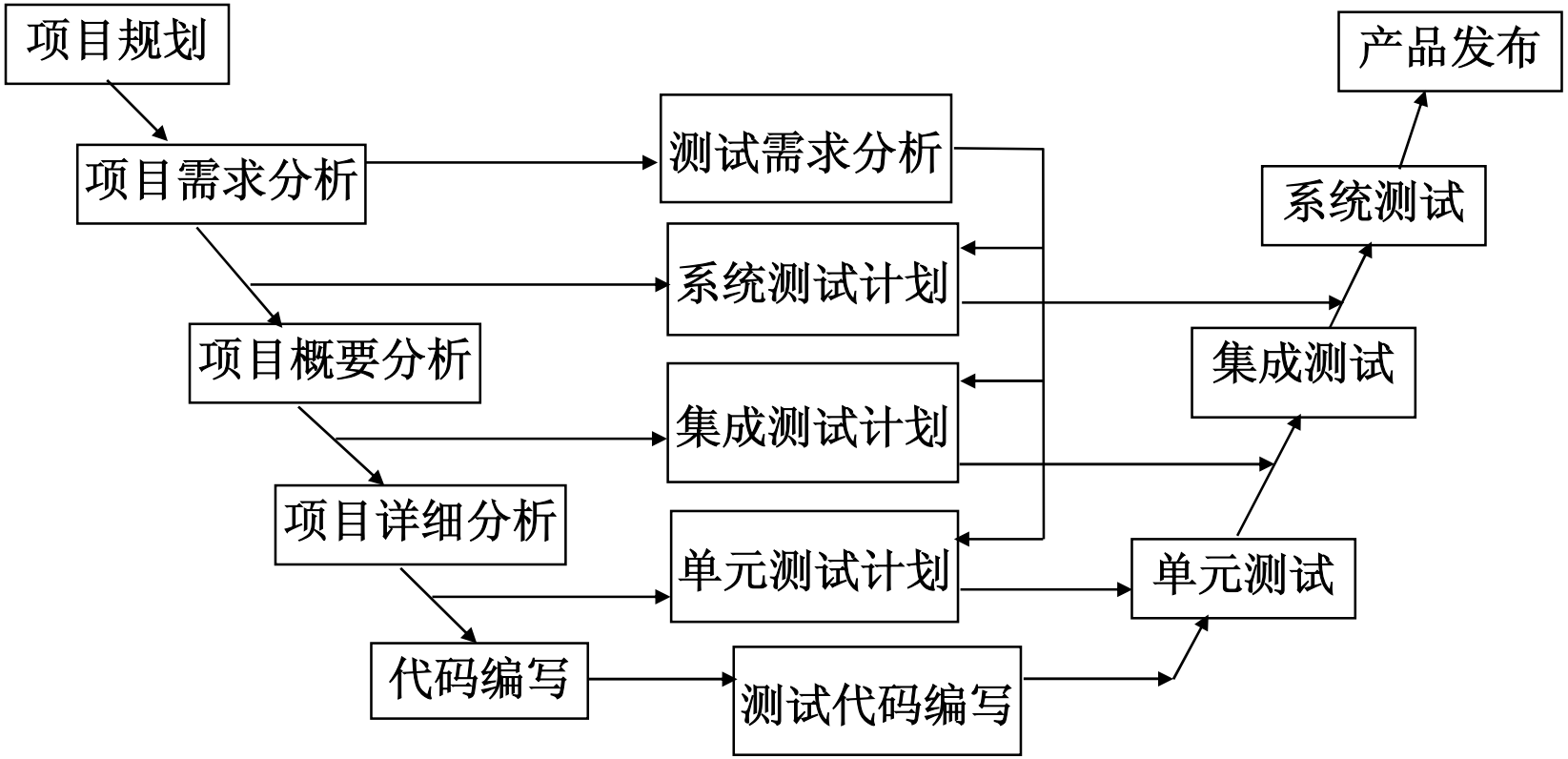


图1-12 完整的开发流程

1.5 软件测试过程

1.5.1 制定测试计划

1.5.2 测试执行过程

Return

1.5.1 制定测试计划

1、制定计划

- 本阶段的主要工作内容
  ——对需求规格说明书的仔细研究
  ——将要测试的产品分解成可独立测试的单元
  ——为每个测试单元确定采用的测试技术
  ——为测试的下一个阶段及其活动制定计划
- 制定计划包括：
  （1）概要测试计划
  （2）详细测试计划

制定测试计划（续）

2、测试大纲（用例）

测试大纲是软件测试的依据，包括测试项目、测试步骤、测试完成的标准。

- 测试大纲的本质

  ——从测试的角度对被测对象的功能和各种特性的细化和展开。

- 测试大纲的好处

  ——保证测试功能不被遗漏，也不被重复测试

  ——合理安排测试人员

  ——使得软件测试不依赖于个人

制定测试计划（续）

3、软件测试报告

软件测试报告是软件测试过程中最重要的文档，它的内容包括：

- 记录问题发生的环境

  ——如：各种资源的配置情况

- 记录问题的再现步骤
- 记录问题性质的说明
- 记录问题的处理进程

  ——问题处理进程从一定角度上反映测试的进程和被测软件的质量状况以及改善过程。

1.5.2 测试执行过程

1、测试执行过程的三个阶段

（1）初测期

——测试主要功能和关键的执行路径，排除主要障碍。

（2）细测期

——依据测试计划和测试大纲、测试用例，逐一测试大大小小的功能、方方面面的特性、性能、用户界面、兼容性、可用性等等；预期可发现大量不同性质、不同严重程度的错误和问题。

（3）回归测试期

——系统已达到稳定，在一轮测试中发现的错误已十分有限；复查已知错误的纠正情况，确认未引发任何新的错误时，终结回归测试。

测试执行过程（续）

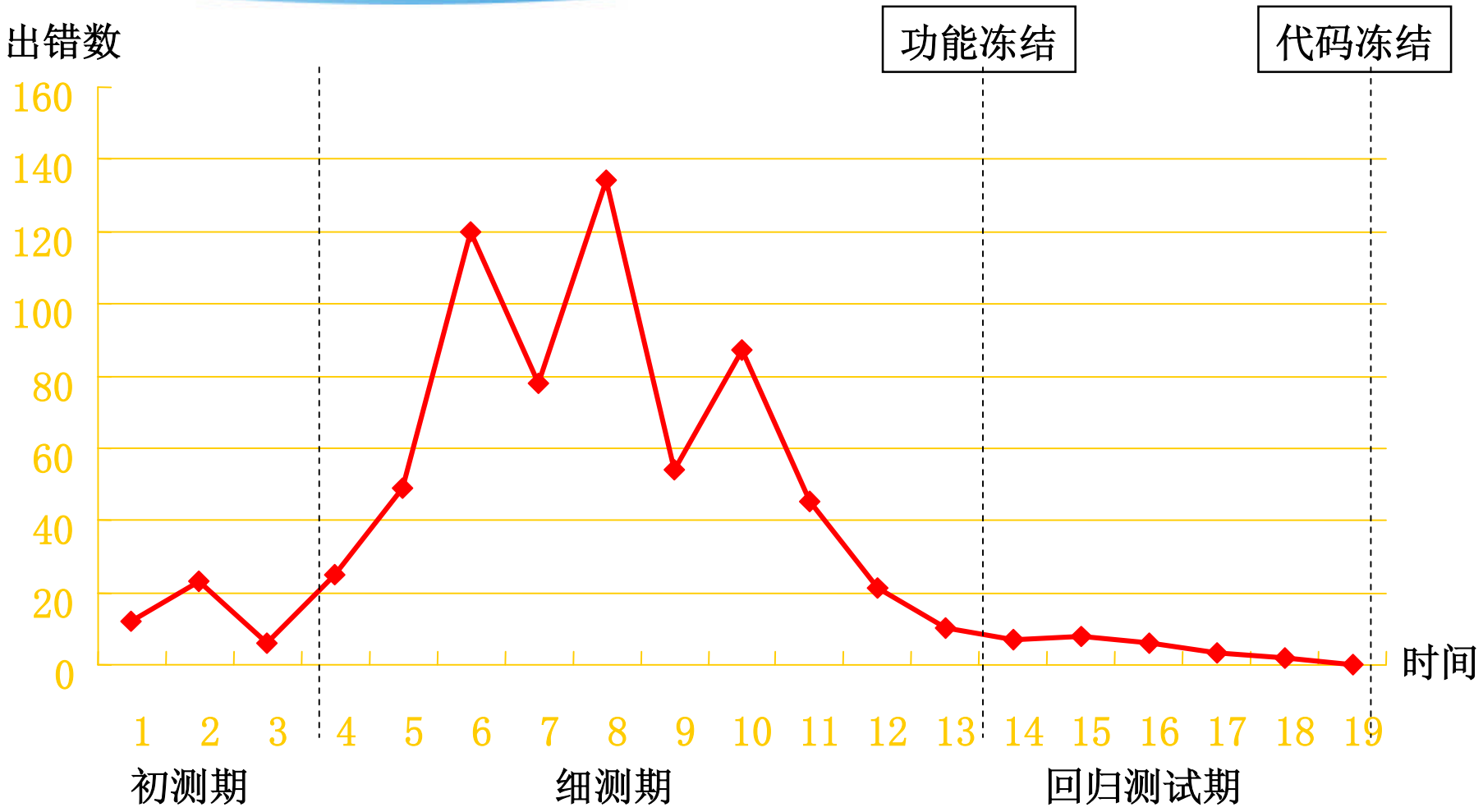


图1-13 三个测试期阶段图示

测试执行过程（续）

2、集成测试过程中的两个重要里程碑

在集成测试过程中的两个重要的里程碑是功能冻结和代码冻结的确定。这两个里程碑界定出回归测试期的起止界限。

- 功能冻结(Function/Feature Freeze)

——经过测试，符合设计要求，确认系统功能和其他特性均不再做任何改变。

- 代码冻结(Code Freeze)

——理论上，在无错误时冻结程序代码，但实际上，代码冻结只标志系统的当前版本的质量已达到预期的要求，冻结程序的源代码，不再对其做任何修改。这个里程碑是设置在软件通过最终回归测试之后。

Thank You !