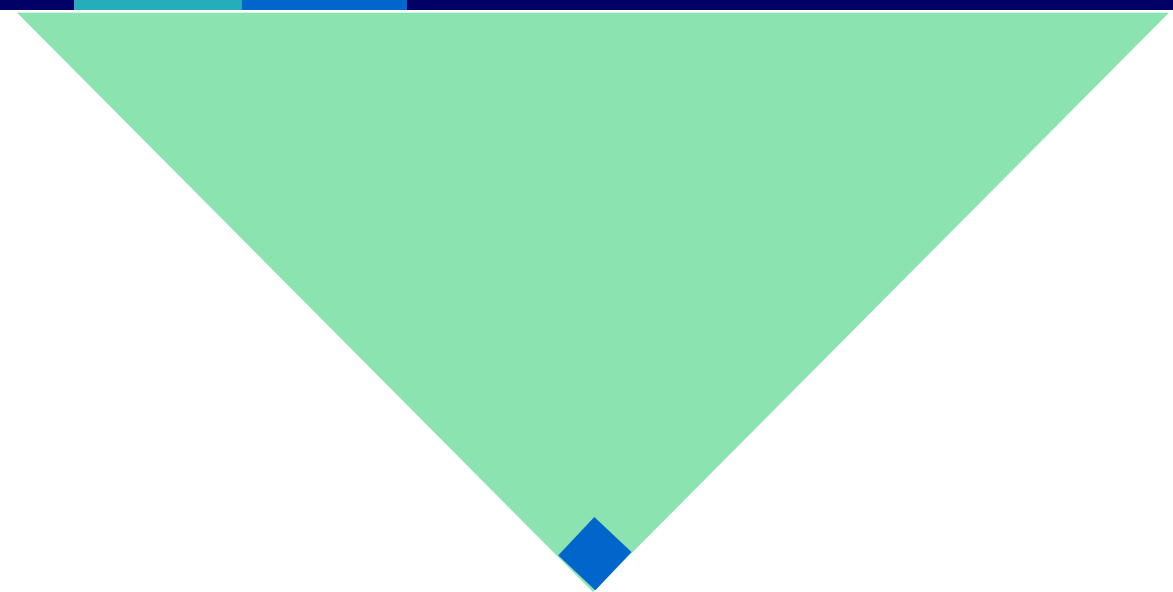




软件测试



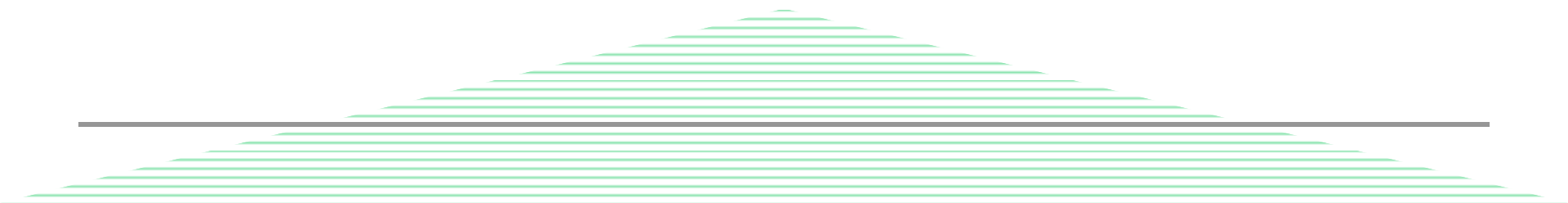
几个问题



学情了解

专业兴趣了解

性格了解





- 课程性质:必修
- 考核: 50%平时+课程设计50%
- 平时: 随堂作业+随堂提问+考勤
- 课程与考证 软件评测师 (软考
www.rkb.gov.cn)
- 课程技能拓展网站 www.51testing.net



几个问题



一、什么是软件？

软件=程序+文档+数据（库）



二、软件的分类

1、按功能分

系统软件

应用软件



二、软件的分类

2、按技术架构分

单机版软件

C/S结构软件

B/S结构软件



二、软件的分类

3、按用户分

产品软件

项目软件



二、软件的分类

4、按开发的规模分

小型——10人以下——1-4个月

中型——10-100人——1年以下

大型——100人以上——1年以上



第1章 软件测试概述



目录



1

软件测试的历史

2

软件缺陷的定义

3

软件缺陷案例

4

软件测试模型

5

软件工程与软件测试的关系

软件缺陷的起源



故事发生在1945年9月9日，一个炎热的下午。当时的机房是一间第一次世界大战时建造的老建筑，没有空调，所有窗户都敞开着。**Grace Hopper**（第一个程序语言编译器的开发人员，后来成为海军少将）正领导着一个研究小组夜以继日地工作，研制一台称为“**MARK II**”的计算机，它使用了大量的继电器（电子机械装置，那时还没有使用晶体管），一台不是纯粹的电子计算机。突然，**MARK II**死机了

板子F第70号继电器中间躺在一只死飞蛾.....



Bug的由来



92

9/9

0800 Antan started
1000 " stopped - antan ✓
1300 (032) MP-MC ~~1.582647000~~ 2.130476415 (3) 4.615925059 (-2)
(033) PRO 2 2.130476415
convd 2.130676415
Relays 6-2 in 033 failed special speed test
in Relay " 10,000 test.

1100 Relays changed
Started Cosine Tape (Sine check)
1525 Started Multi-Adder Test.

1545



Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

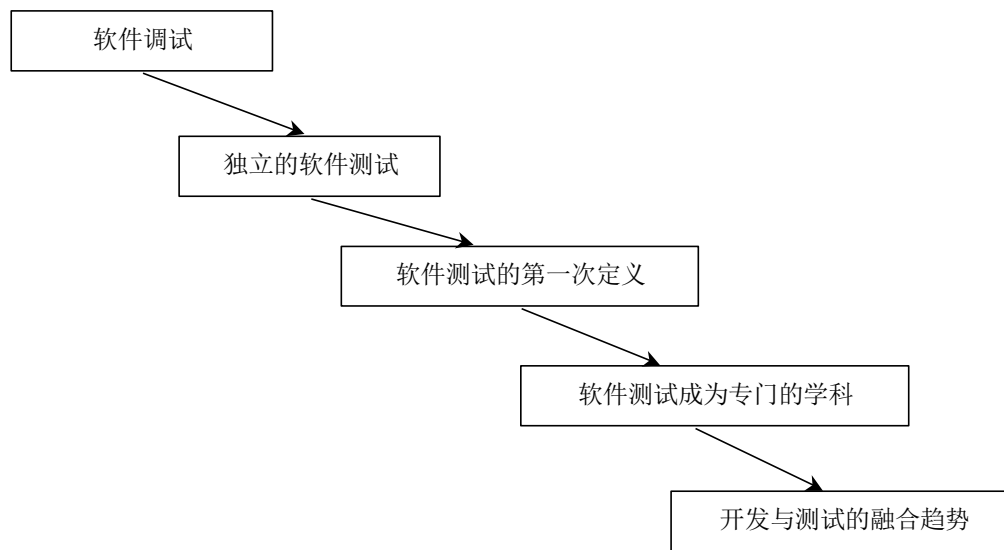
1630 Antan started.

1700 closed down.

Grace Hopper小心地用镊子将飞蛾夹出来，用透明胶布贴到“事件记录本”

中，并注明“第一个发现虫子的实例”。

软件测试的发展



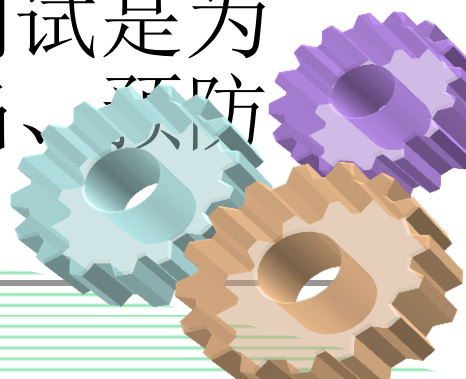


软件测试学科的发展



从测试的思想导向来划分为4个阶段：

- 1957～1978年，以功能验证为导向，测试是证明软件是正确的（正向思维）。
- 1978～1983年，以破坏性为为导向，测试是为了找到软件中的错误（逆向思维）。
- 1983～1987年，以质量评估为导向，测试是提供产品的评估和质量度量。
- 1988年起，以缺陷预防为导向，测试是为了展示软件符合设计要求，发现缺陷、预防缺陷。





两种观点



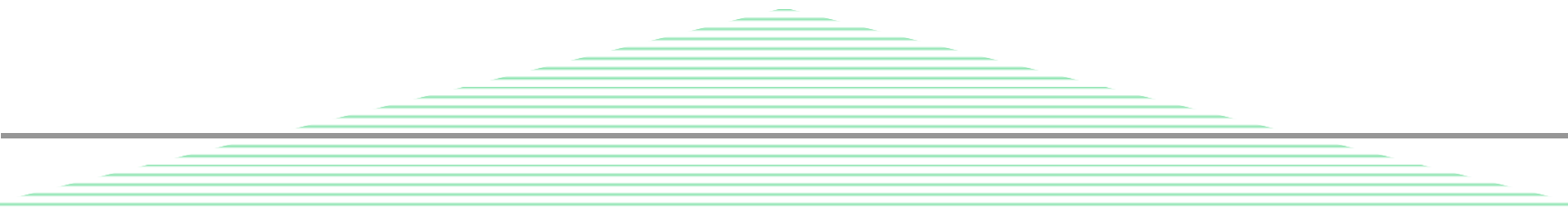
- 测试是为了证明程序无错误;
- 测试是为了证明程序有错;
- **G. Myers**提出软件测试是为了发现错误而执行程序的过程。
- **Bill Hetzel**提出了测试的目的不仅仅是为了发现软件缺陷与错误，而且也是对软件质量进行度量和评估，以提高软件的质量。



软件测试的正面性（1）



Bill Hetzel博士（正向思维的代表）：

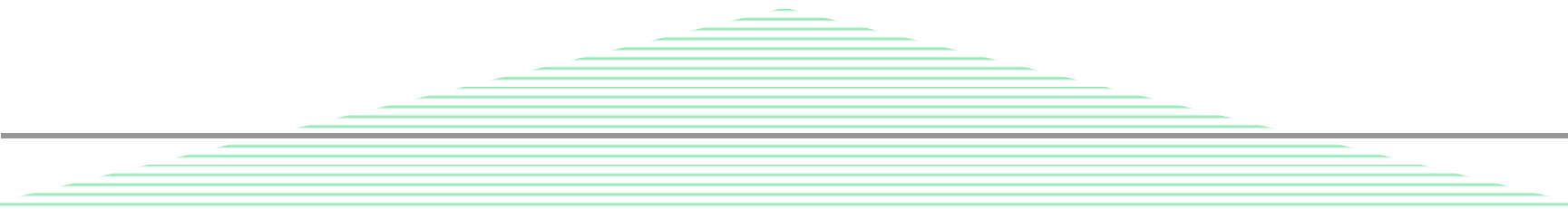
- 软件测试就是为程序能够按预期设想那样运行而建立足够的信心。
 - “软件测试是一系列活动以评价一个程序或系统的特性或能力并确定是否达到预期的结果”
 - 测试是为了验证软件是否符合用户需求，即验证软件产品是否能正常工作
- 



软件测试的正面性（2）



IEEE 的定义：

- 在特定的条件下运行系统或构件，观察或记录结果，对系统的某个方面做出评价
 - 分析某个软件项以发现现存的和要求的条件之差别（即错误）并评价此软件项的特性
- 



软件测试的反面性



Glenford J. Myers（反向思维的代表）：

- ▣ 测试是为了证明程序有错，而不是证明程序无错误
- ▣ 一个好的测试用例是在于它能发现至今未发现的错误
- ▣ 一个成功的测试是发现了至今未发现的错误的测试



软件测试定义的两面性



软件测试

正向思维—
验证软件正常工作

评价一个程序或系统的特性或能力并确定是否达到预期的结果

在设计规定的环境下运行软件的所有功能，直至全部通过。

逆向思维—
假定软件有错误

测试是为发现错误而针对某个程序或系统的执行过程

寻找容易犯错误的地方和系统的薄弱环节，试图破坏系统，直至找不出问题。



软件测试定义？



■ 常见概念

- ✧ 保证程序和相应的规范说明一致。
- ✧ 发现软件中的缺陷。
- ✧ 确保软件不做不必要的事情。
- ✧ 确保系统合理地执行。
- ✧ 确保系统失败前可以让系统运行到何种程度
- ✧ 确保发布给用户的系统中有哪些风险。
- ✧ ISO9000定义：测试是一种基于机器的，对代码执行测试，确认测试的活动。



1983年IEEE (Institute of Electrical and Electronics Engineers) 国际电子电气工程师协会提出的软件工程标准术语中给软件测试下的定义是：使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别。

我们把软件测试定义为在程序中找到故障的过程，使测试成为可以做到的任务，从而克服了心理上存在的问题。因此，对软件测试人员而言，测试的最好定义是：软件测试是为了发现错误而执行程序的过程。



计算器的例子



- 计算器说明书：该计算器将准确无误地进行加、减、乘、除运算。计算器不会出现崩溃、死锁或停止反应。
 - (1) $2+3$ ，没有反应？
 - (2) 随意敲击键盘后，没有了反应？
 - (3) 还能计算某数的平方根
 - (4) 因为电池没有电，所以计算错了
 - (5) 按键很小、显示屏看不清楚



软件缺陷



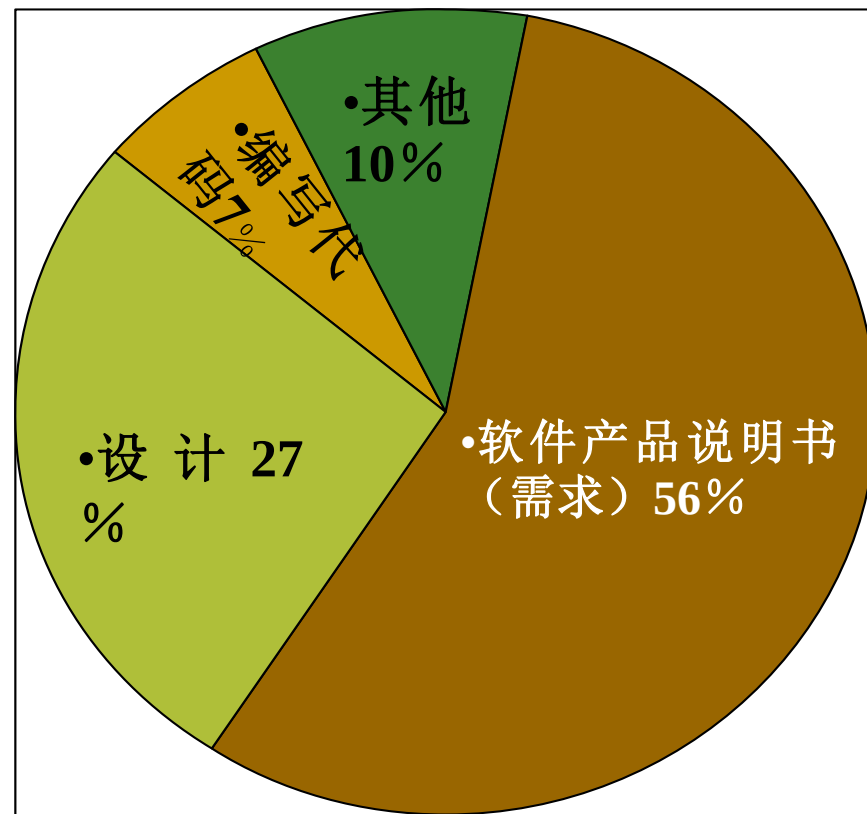
- **1、什么是软件缺陷？**
 - (1)** 软件未达到产品说明书中已经标明的功能；
 - (2)** 软件出现了产品说明书中指明不会出现的错误；
 - (3)** 软件未达到产品说明书中虽未指出但应当达到的目标；
 - (4)** 软件功能超出了产品说明书中指明的范围；
 - (5)** 软件测试人员认为软件难以理解、不易使用，或者最终用户认为该软件使用效果不良。

2、为什么会产生软件缺陷？



- (1) 产品需求说明书
- (2) 设计方案
- (3) 编写代码
- (4) 其他

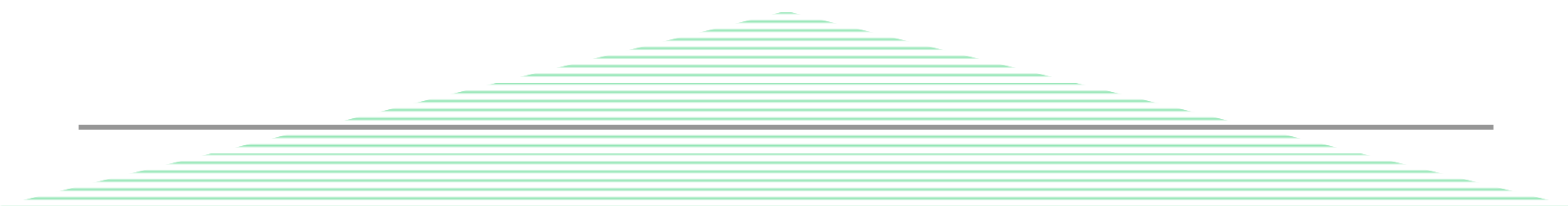
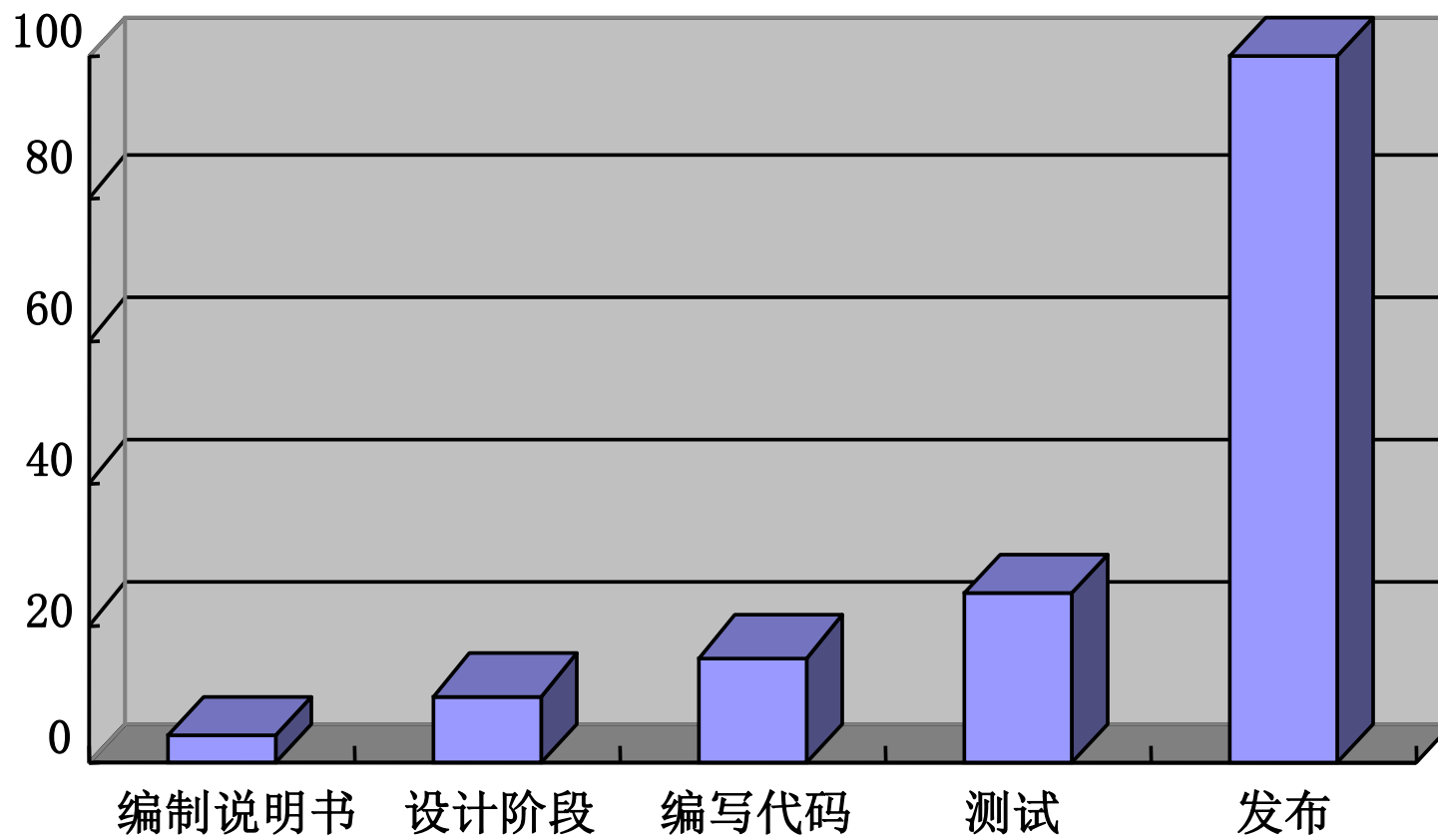
图1-1 软件缺陷产生的原因分布



3、软件缺陷修复的费用

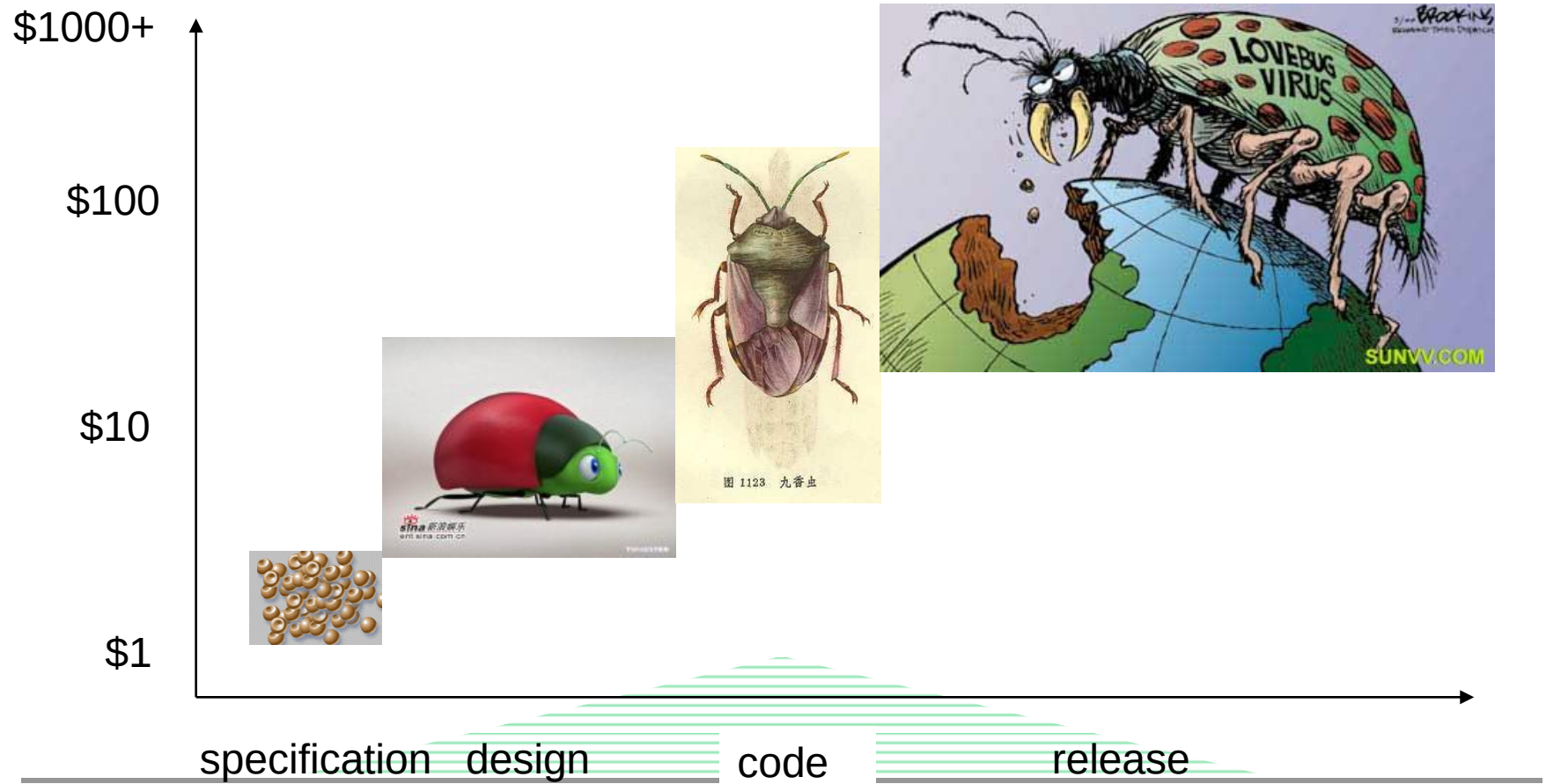


- 软件在从需求、设计、编码、测试一直到交付用户公开使用后的过程中，都有可能产生和发现缺陷。随着整个开发过程的时间推移，更正缺陷或修复问题的费用呈几何级数增长。





- The cost are logarithmic





软件缺陷例子1



■ 事件

✧ 1996年6月4日，Arane5发射40秒后爆炸。

■ 原因

✧ 将一个64位浮点值转换为16位有符号整数值时，超出了16位整数的表示范围，而这个异常未得到正确解决



软件缺陷例子2



■ 事件

- ◇ 1999年；
- ◇ 火星气象卫星 (Mars Climate Orbiter) 到达火星之后不久就消失；
- ◇ 火星极地登陆者 (Mars Polar Lander) 在火星上着陆时坠毁



■ 原因

- ◇ 地面系统软件和飞行器上软件分别使用公制和英制两种单位。



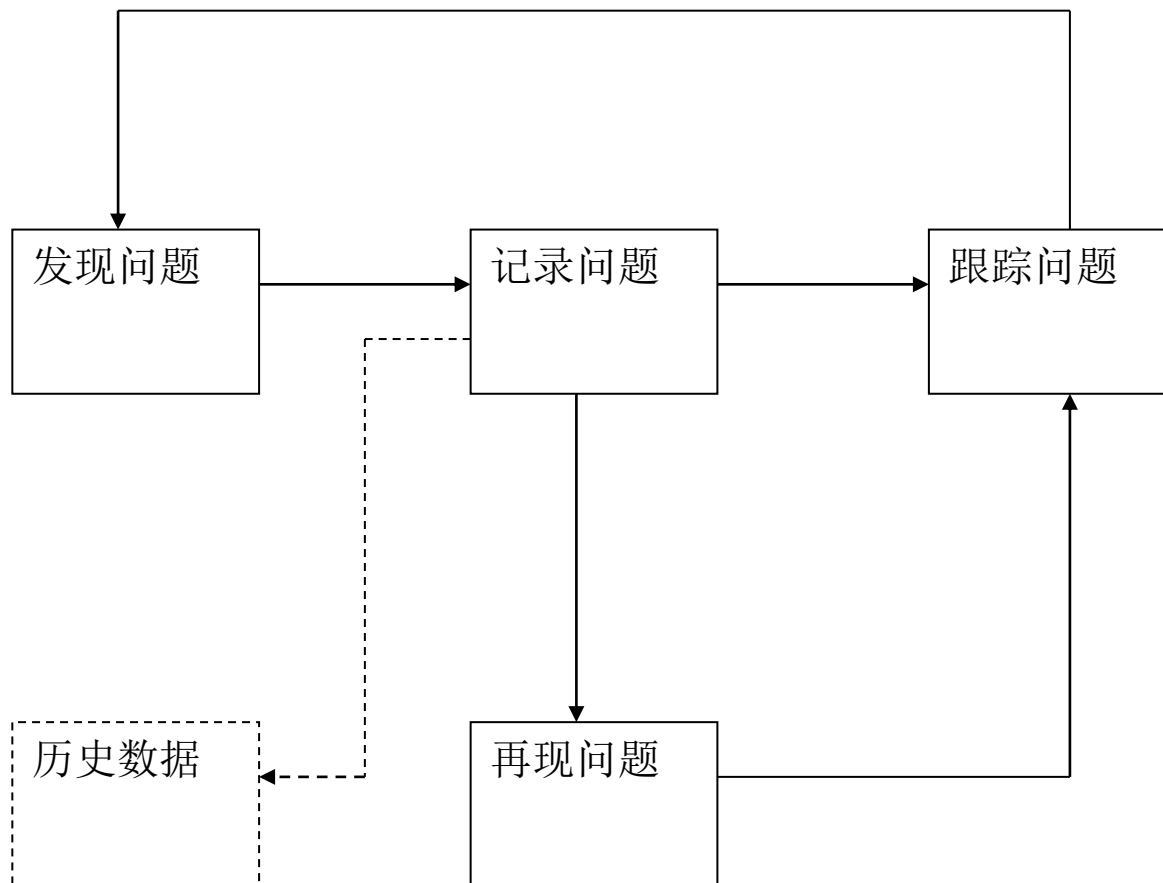
- 神舟七号:
- 测试用例**3**万多。确保稳定，确保万无一失。
- 基本消除缺陷



软件测试定义

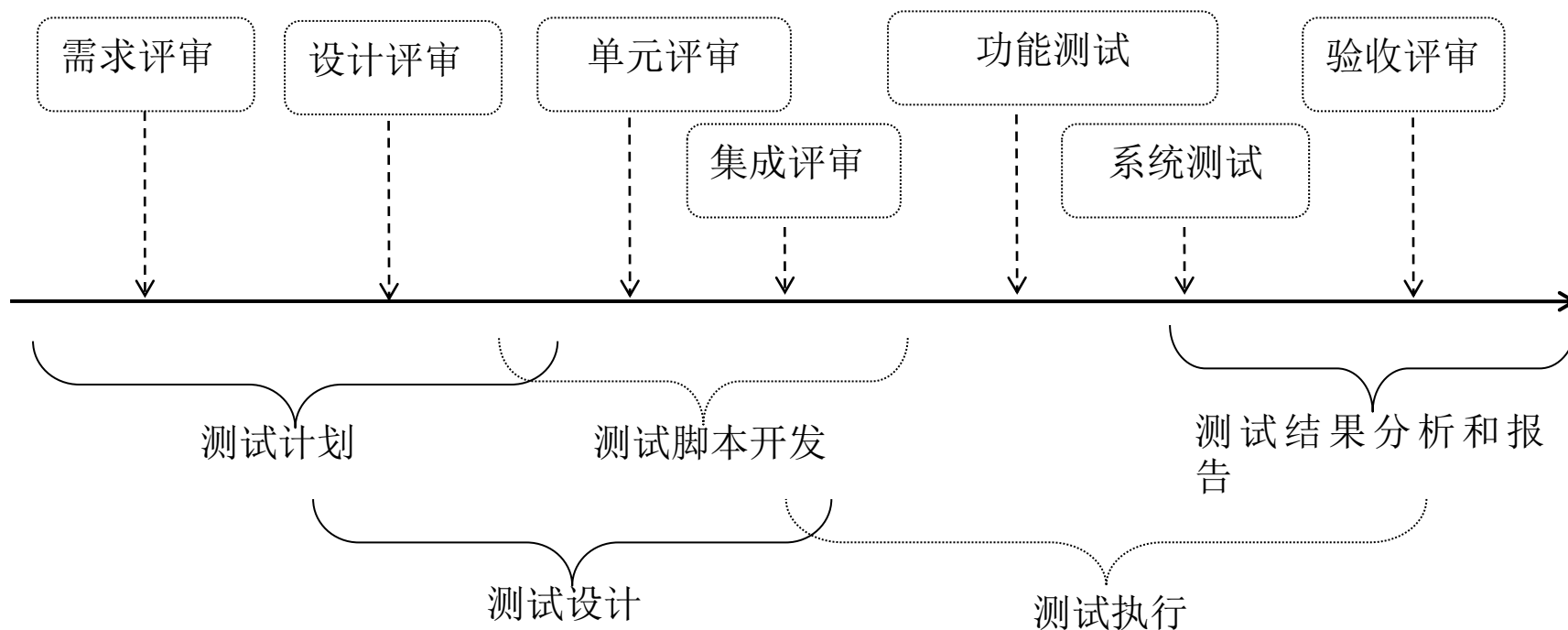
■ 完整定义

◇ 软件测试是发现并指出软件（包含软件经过建模、需求、设计等阶段所产生的大量输出工件及程序代码）中存在缺陷的过程，这个过程指明和标注问题存在的正确位置，详细记录导致问题出现的操作步骤，及时存储当时的错误状态，以上组合在一起便于测试后问题能够准确再现。





软件测试过程





软件测试阶段



阶 段	输 入	输 出
需求分析	需求定义, 市场分析文档, 相关技术文档	市场需求分析会议纪要, 功能设计, 技术设计
设计审查	市场需求文档, 技术设计文档	测试计划, 测试用例
功能验证	代码完成文件包, 功能详细设计说明书 最终技术文档	完整测试用例, 完备的测试计划, 缺陷报告, 功能验证测试报告
系统测试	代码修改后的文件包 完整测试用例, 完备的测试计划	缺陷报告 缺陷状态报告 项目阶段报告
确认测试	代码冻结文件包 确认测试用例	缺陷状态报告 缺陷报告审查 版本审查
版本发布	代码发布文件包 测试计划检查清单	当前版本已知问题的清单 版本发布报告



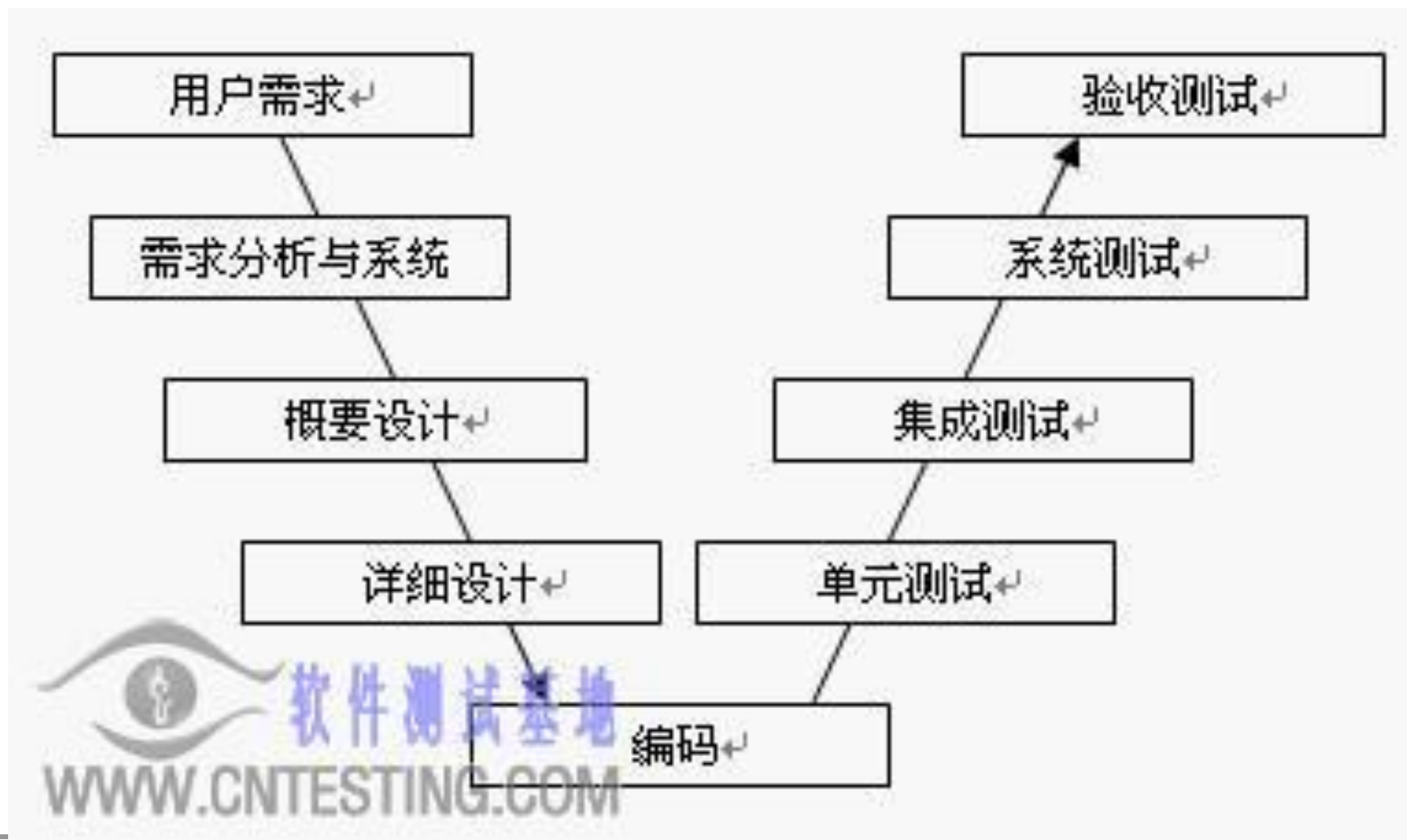
软件测试模型



- 零售店里的服装模特是一个模型
- 会计的公式： $\text{资产} = \text{债务} + \text{所有者权益}$ 也是一个模型
- 模型是对现实的抽象概括，对某些真实事件的简化表征。



“V”模型





“V”模型



- **V模型**: 20世纪80年代后期,**Paul Rook**提出了著名的软件测试的**V模型**, 是瀑布模型的变种, 它反映了测试活动与分析和设计的关系, 非常明确的表明了测试过程中存在的不同级别, 以及各测试阶段与开发过程中的各阶段的对应关系, 图中的箭头代表了时间方向, 左边下降的是开发各阶段, 右边上升的是测试过程的各个阶段。



“V”模型



- V模型指出：
 - 单元测试和集成测试应检测程序的执行是否满足软件设计的要求;
 - 系统测试应检测系统功能,性能的质量特性是否达到系统要求的指标;
 - 验收测试确定软件的实现是否满足用户需要或合同的要求.



“V”模型



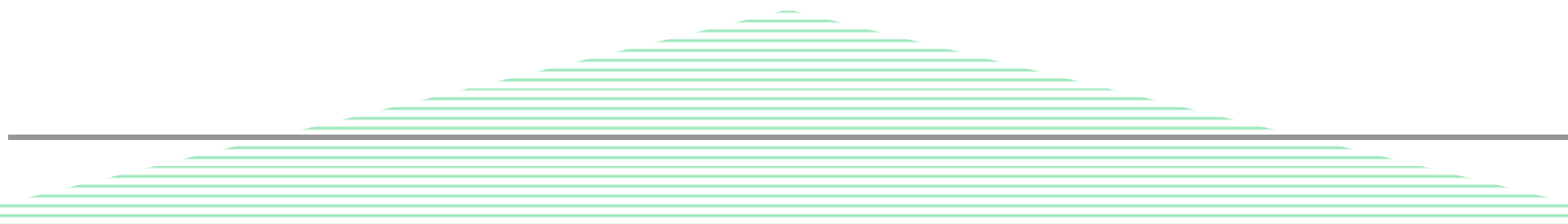
- V模型的缺陷
 - 仅仅把测试过程作为在需求分析、系统设计及编码之后的一个阶段
 - 忽视了测试对需求分析,系统设计的验证,一直到后期的验收测试才被发现。



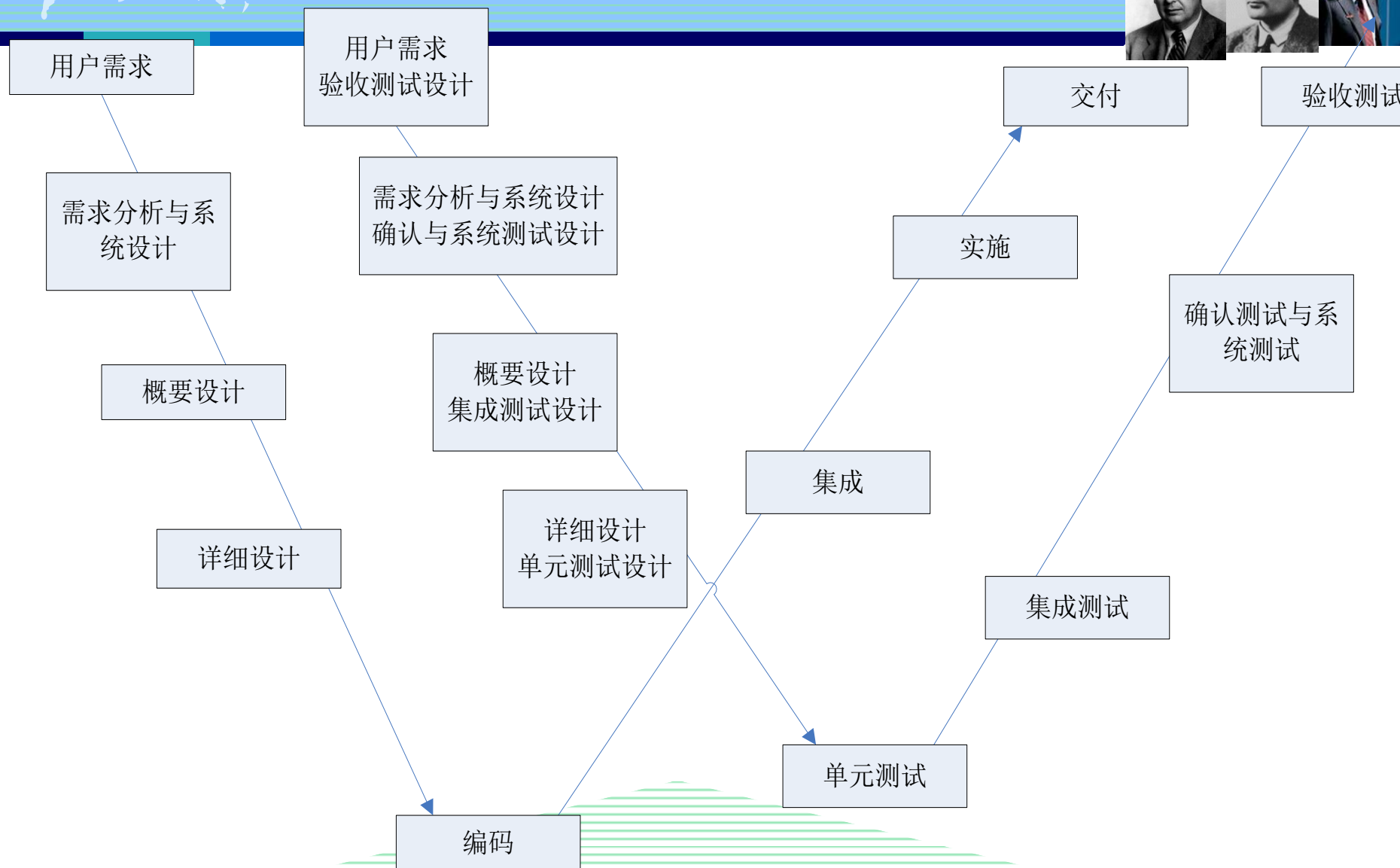
“W”模型



- **Evolutif**公司提出了**W**模型的概念，增加了软件各开发阶段中应**同步进行的验证和确认**活动,明确了测试与开发的并行性.



“W”模型





W模型



- 测试伴随着整个软件开发周期
- 测试的对象不仅仅是程序，需求、设计和功能同样要测试
- 根据W模型的要求，一旦有文档提供，就要及时确定测试的条件、编写测试用例



W模型的局限性



- 在W模型中，需求、设计、编码等活动被视为串行的，同时，测试和开发活动也保持着一种线性的前后关系，上一阶段完全结束，才可正式开始下一个阶段工作。
- 无法支持迭代、自发性以及变更调整。

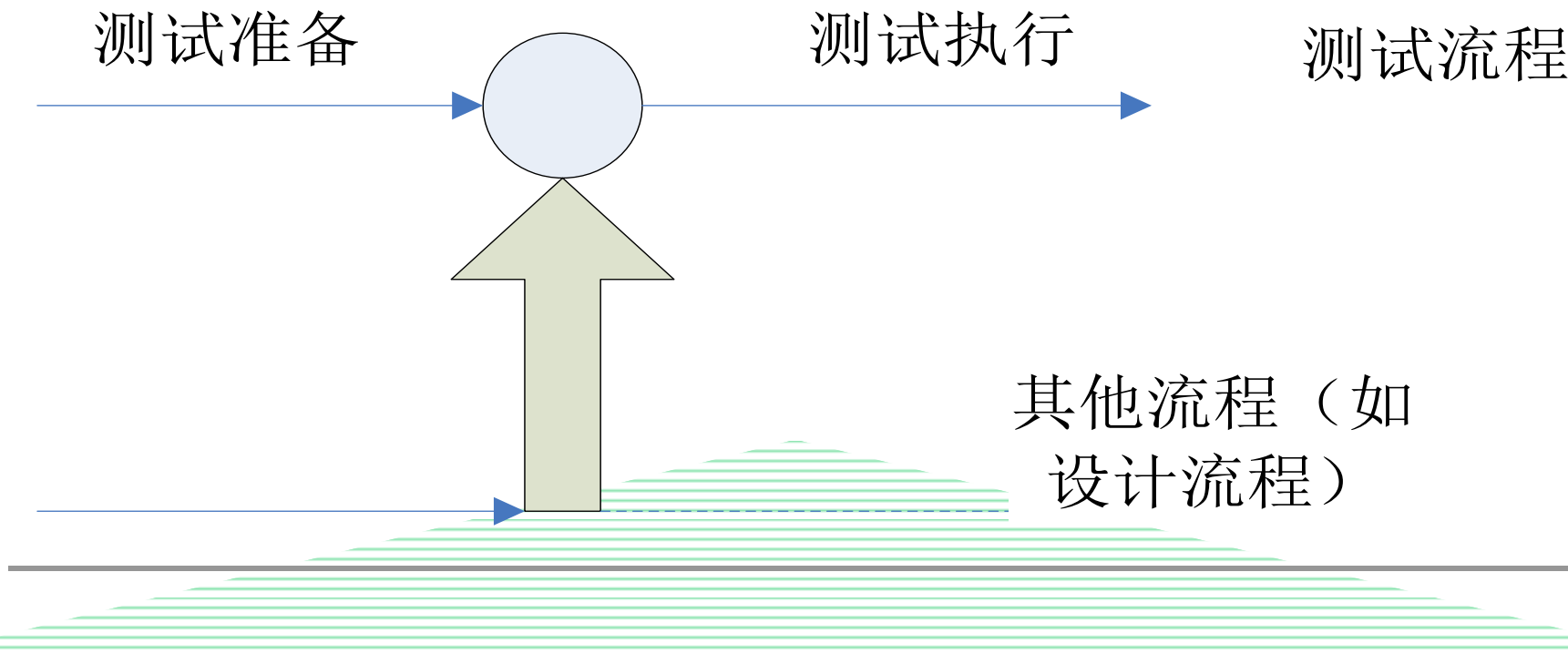


H模型



- 这个示意图仅仅演示了在整个生产周期中某个层次上的一次测试“微循环”。图中的其他流程可以是任意开发流程。

测试就绪点



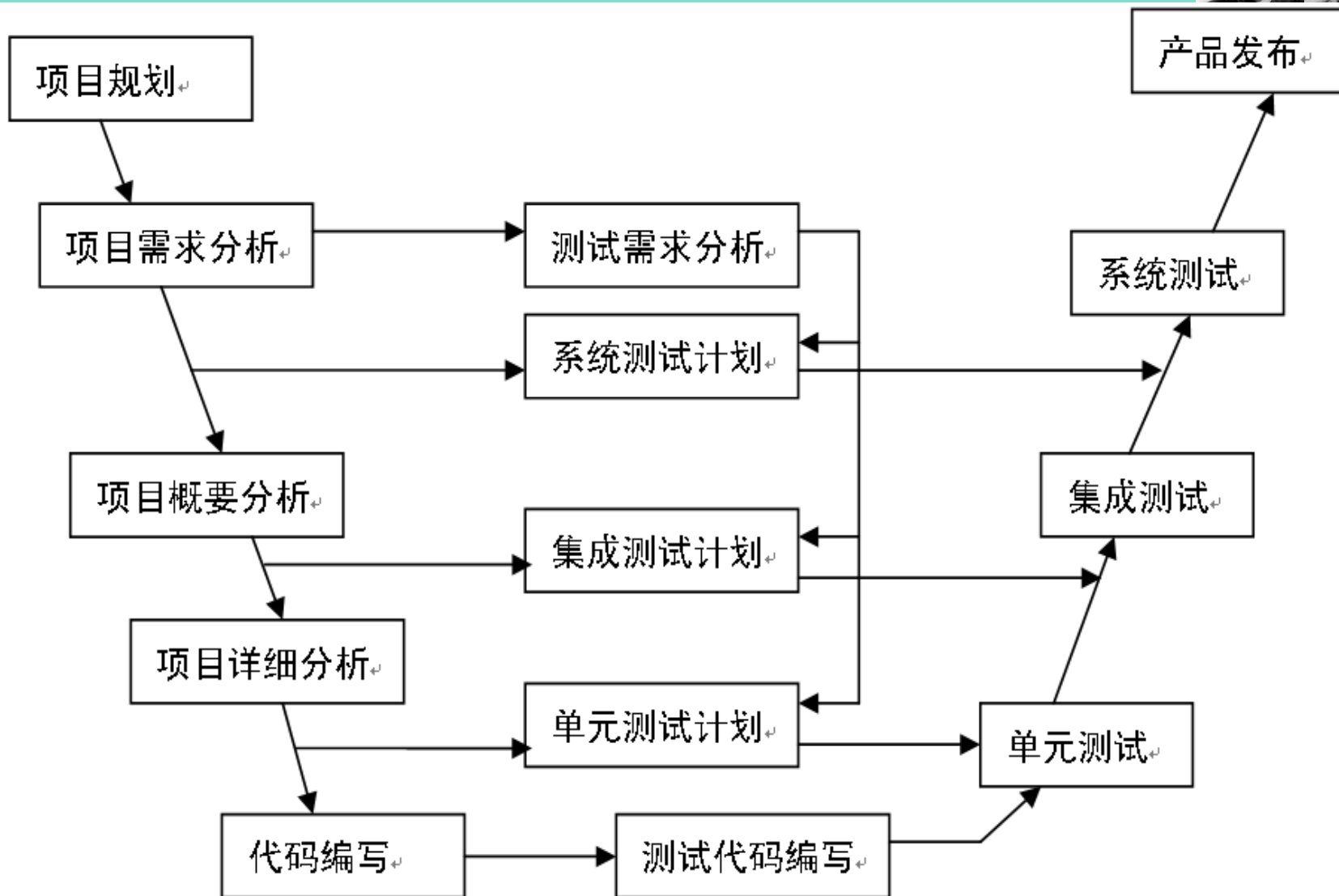


H模型



- 软件测试不仅仅指测试的执行，还包括很多其他的活动。
- 软件测试是一个独立的流程，贯穿产品整个生命周期，与其他流程并发的进行。
- 软件测试要尽早准备，尽早执行。
- 软件测试是根据被测物的不同而分层进行的。不同层次的测试活动可以按照某个次序先后进行的，但也可能是反复的。

软件工程与软件测试的关系



软件工程与软件测试的关系



测试在开发阶段的作用如下：

- (1) 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。
- (2) 需求分析阶段：确定测试需求分析、系统测试计划的制定。其中，测试需求分析是对产品生命周期中测试所需求的资源、配置、每阶段评判通过的规约；系统测试计划则是依据软件的需求规格说明书，制定测试计划 and 设计相应的测试用例。
- (3) 概要设计和详细设计阶段：确保集成测试计划和单元测试计划完成。
- (4) 编码阶段：由开发人员进行自己负责部分的测试代码。



思考题



- 复习——通过课后思考题进行本节课的知识回顾与巩固
- P18



Thank You !

