SOFTWARE TESTING

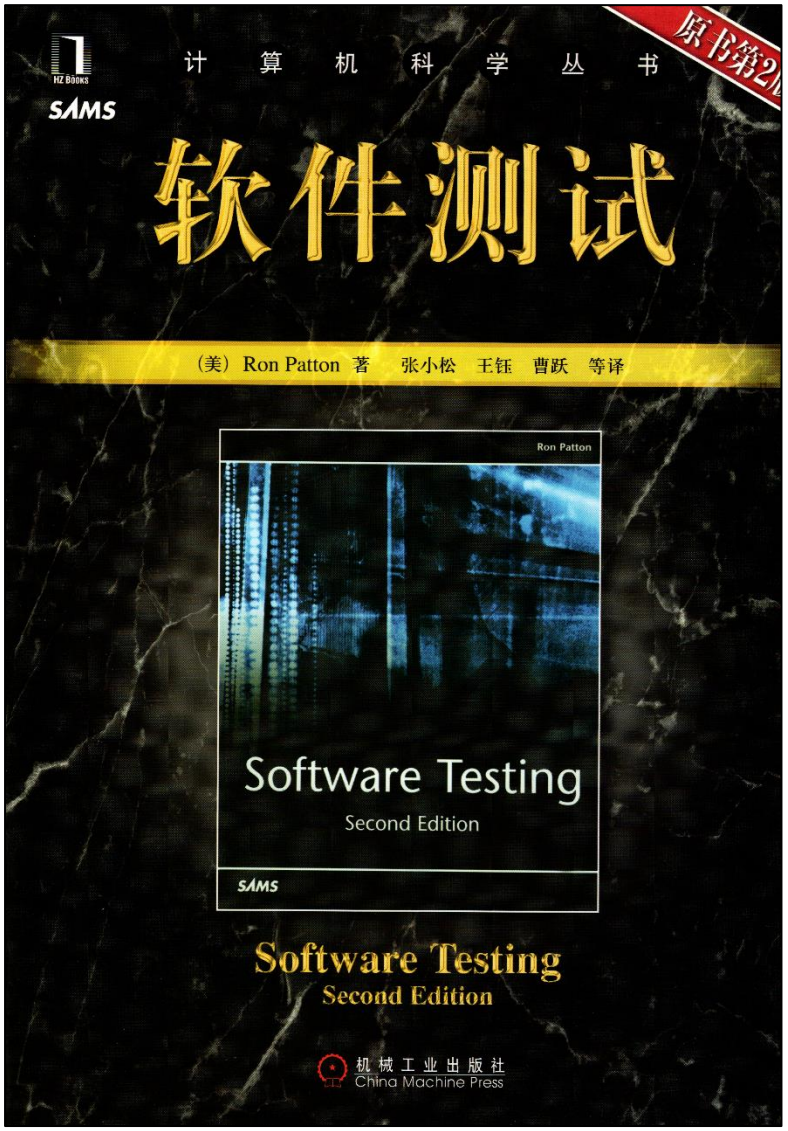
苏临之

sulinzhi029@nwu.edu.cn

Preparations

Something that you should master in advance:

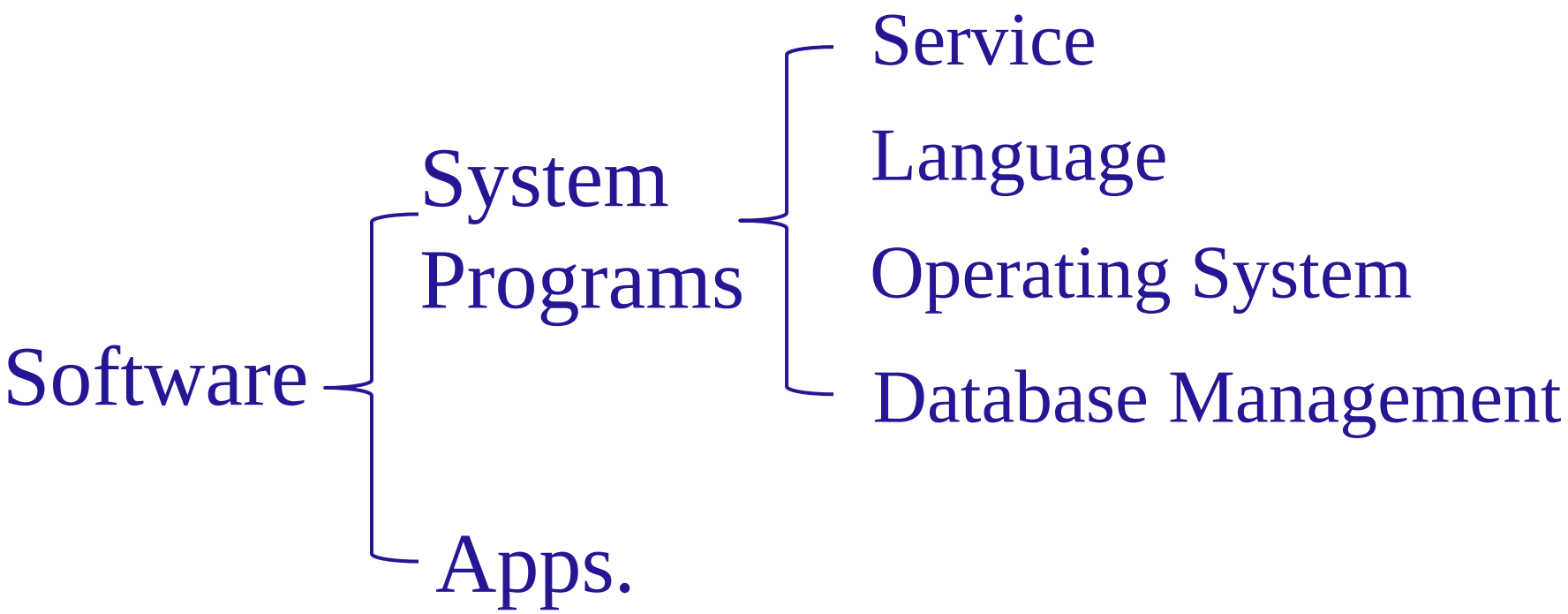
1. Software Engineering
2. Programming & Algorithm
3. Java




Nikon
NIKKOR 18X OPTICAL ZOOM VR
18x


08:08
Tue, Mar 27

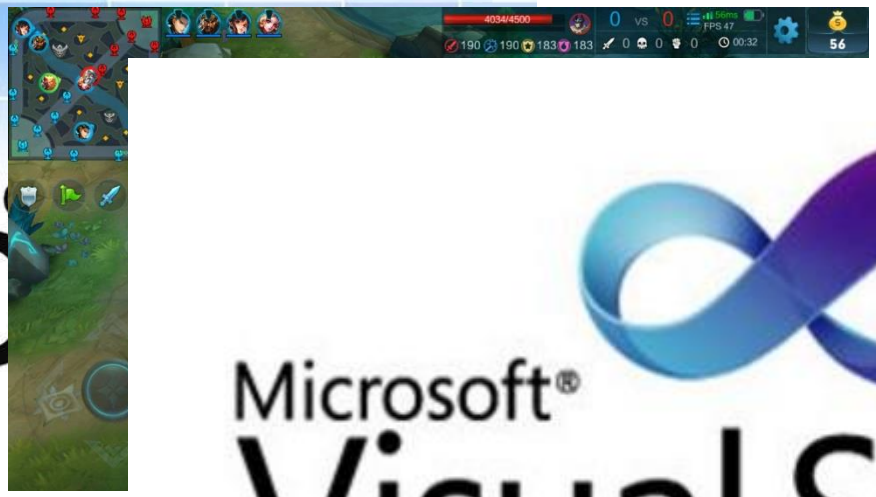
Software in a Computer



Software & Hardware

- Dependency: The hardware serves as the base for the software, whereas the software shows out the potential of the hardware.
- Unboundedness: Some functions can be achieved by both software and hardware.
- Cooperativity: The software and hardware can not be developed dependently.

What Software Have You Used?

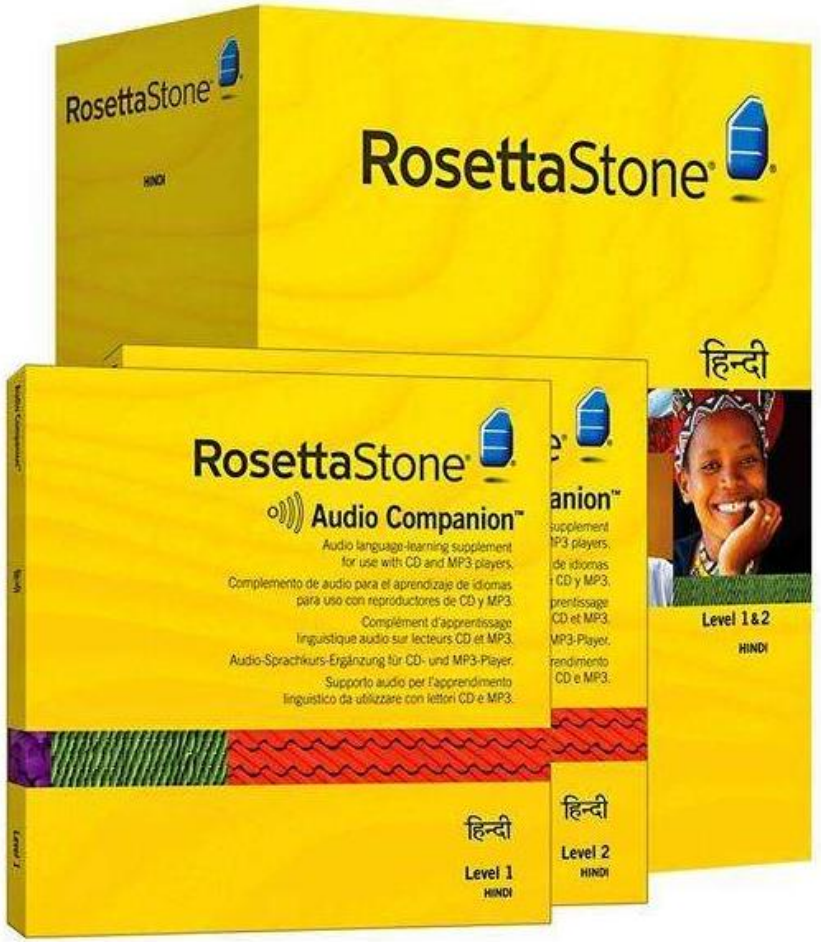


Mi
C

Microsoft®
Visual Studio®

What Comprises the Software?

- 帮助文档
- 用户手册
- 样品和示例
- 标签和不干胶
- 产品支持信息
- 图标和标志
- 错误信息
- 广告和宣传材料
- 安装
- 说明文件



Software Project Members

- 项目经理
- 需求人员
- 体系架构师/系统工程师
- 程序员、开发人员
- 测试员或质量保证员(Quality Assurance, QA)
- 手册编写者
- 配置管理员

Background for Software Testing

- It's easy to take software for granted and not really appreciate how much it has infiltrated our daily lives.
- Software is everywhere.
  - Video games
  - Internet
  - Long-distance phone service
  - Medical treatments
  - ……
- Software is written by people so it's not perfect.

Error, Failure, Fault and Defect

- Error: The software can not operate as fully anticipated.
- Failure: The software engenders wrong results caused by errors.
- Fault: The existing physical problems that may or may not result in failures.
- Defect: The software does not work as required. A software defect, or bug as often referred to, may or may not result in an error.

Identification of a Software Defect

- A software defect occurs when one or more of the following five rules is true:
  1. The software doesn't do something that the product specification says it should do.
  2. The software does something that the product specification says it shouldn't do.
  3. The software does something that the product specification doesn't mention.
  4. The software doesn't do something that the product specification doesn't mention but should.
  5. The software is difficult to understand, hard to use, slow, or in the tester's eyes will be viewed not good by the end user.

A Simple Example

- Show an array 'a' where a[i]=i, ('i' stands for the integer), and then output it reversely. (Using CPL)

- 代码中没有声明变量类型，因此系统无法预先分配变量空间，导致整数i、k和数组a无法被系统识别，因此是软件故障，会导致软件失效。

```
#include <stdio.h>
int main(void){
    for(i=0;i<=9;i++)
        a[i]=i;
    for(k=9;k>0;k--)
        printf("%d ",a[k]);
    return 0;
}
```

A Simple Example

- Show an array 'a' where a[i]=i, ('i' stands for the integer), and then output it reversely. (Using CPL)

- 代码运行无误，但是第二个for循环中k>0这一项会导致少输出一个数，因此这是软件故障。

```
#include <stdio.h>
int main(void){
    int i,k,a[10];
    for(i=0;i<=9;i++)
        a[i]=i;
    for(k=9;k>0;k--)
        printf("%d ",a[k]);
    return 0;
}
```

A Simple Example

- Show an array 'a' where a[i]=i, ('i' stands for the integer), and then output it reversely. (Using CPL)

- 代码运行无误，基本功能也已经实现。但循环变量用了i和k，多占了内存空间，会浪费存储空间，因此这是一个小型软件缺陷。

```
#include <stdio.h>
int main(void){
    int i,k,a[10];
    for(i=0;i<=9;i++)
        a[i]=i;
    for(k=9;k>=0;k--)
        printf("%d ",a[k]);
    return 0;
}
```

A Simple Example

- Show an array 'a' where a[i]=i, ('i' stands for the integer), and then output it reversely. (Using CPL)

- 最终代码统一使用循环变量i，消除了刚才的诸多错误和缺陷。

```
#include <stdio.h>
int main(void){
    int i,a[10];
    for(i=0;i<=9;i++)
        a[i]=i;
    for(i=9;i>=0;i--)
        printf("%d ",a[i]);
    return 0;
}
```

Infamous Software Defect Examples

- The US Patriot Missile Event
- Intel Pentium Floating-Point Division
- Disney's Lion King
- The Year 2000 (Y2K) Bug
- ……

The US Patriot Missile Event

- The U.S. Patriot missile defense system is a scaled-back version of the Strategic Defense Initiative ("Star Wars") program proposed by President Ronald Reagan. It was first put to use in the Gulf War as a defense for Iraqi Scud missiles.
- Although there were many news stories touting the success of the system, it did fail to defend against several missiles, including one that killed 28 US soldiers.
- A small timing error in the system's clock accumulated to the point that after 14 hours, the tracking system was no longer accurate. In the attack, the system had been operating for more than 100 hours.

Intel Pentium Floating-Point Division

- (4195835 / 3145727) × 3145727 - 4195835
- On October 30, 1994, Dr. Thomas R. Nicely of Lynchburg (Virginia) College traced an unexpected result from one of his experiments to an incorrect answer by a division problem solved on his Pentium PC. He posted his find on the Internet and soon afterward a firestorm erupted as numerous other people duplicated his problem and found additional situations that resulted in wrong answers.
- Fortunately, these cases were rare and resulted in wrong answers only for extremely math-intensive, scientific, and engineering calculations. Most people would never encounter them doing their taxes or running their businesses.

Intel Pentium Floating-Point Division

- Their software test engineers had found the problem while performing their own tests before the chip was released.
- Intel's management decided that the problem wasn't severe enough or likely enough to warrant fixing it or even publicizing it.
- Once the bug was found, Intel attempted to diminish its perceived severity through press releases and public statements. When pressured, Intel offered to replace the faulty chips, but only if a user could prove that he was affected by the bug.

Intel Pentium Floating-Point Division

- There was a public outcry. Internet newsgroups were jammed with irate customers demanding that Intel fix the problem. News stories painted the company as uncaring and incredulous.
- In the end, Intel apologized for the way it handled the bug and took a charge of more than $400 million to cover the costs of replacing bad chips.
- So what makes this story notable isn't the bug, but the way Intel handled the situation.

Disney's Lion King

- In the autumn of 1994, the Disney company released its first multimedia CD-ROM game for children, The Lion King Animated Storybook. However, on December 26Disney's customer support phones began to ring. Soon the phone support technicians were swamped with calls from angry parents with crying children who couldn't get the software to work. Numerous stories appeared in newspapers and on TV news.
- It turns out that Disney failed to test the software on a broad representation of the many different PC models available on the market. The software worked on a few systems likely the ones that the Disney programmers used to create the game but not on the most common systems that the general public had.

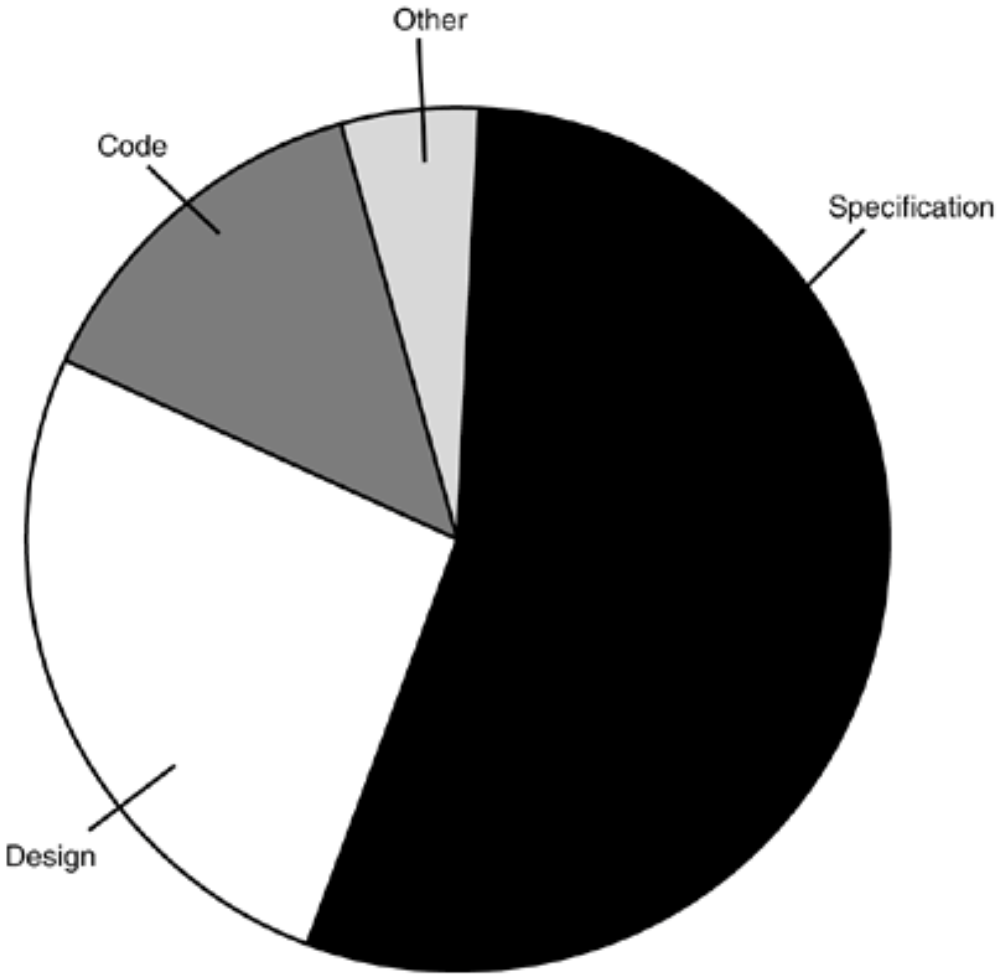
The Year 2000 (Y2K) Bug

- Sometime in the early 1970s a computer was working on a payroll system for his company. The computer he was using had very little memory for storage, forcing him to conserve every last byte he could. One method he used was to shorten dates from their 4-digit format, such as 1973, to a 2-digit format, such as 73.
- He briefly considered the problems that might occur when the current year hit 2000. But he thought that there would have been an update before 2000.
- Unfortunately, in 1995, the system was still used but the one who developed it has been retired. No one knew how to get into the program to check if it was Y2K compliant, let alone how to fix it.

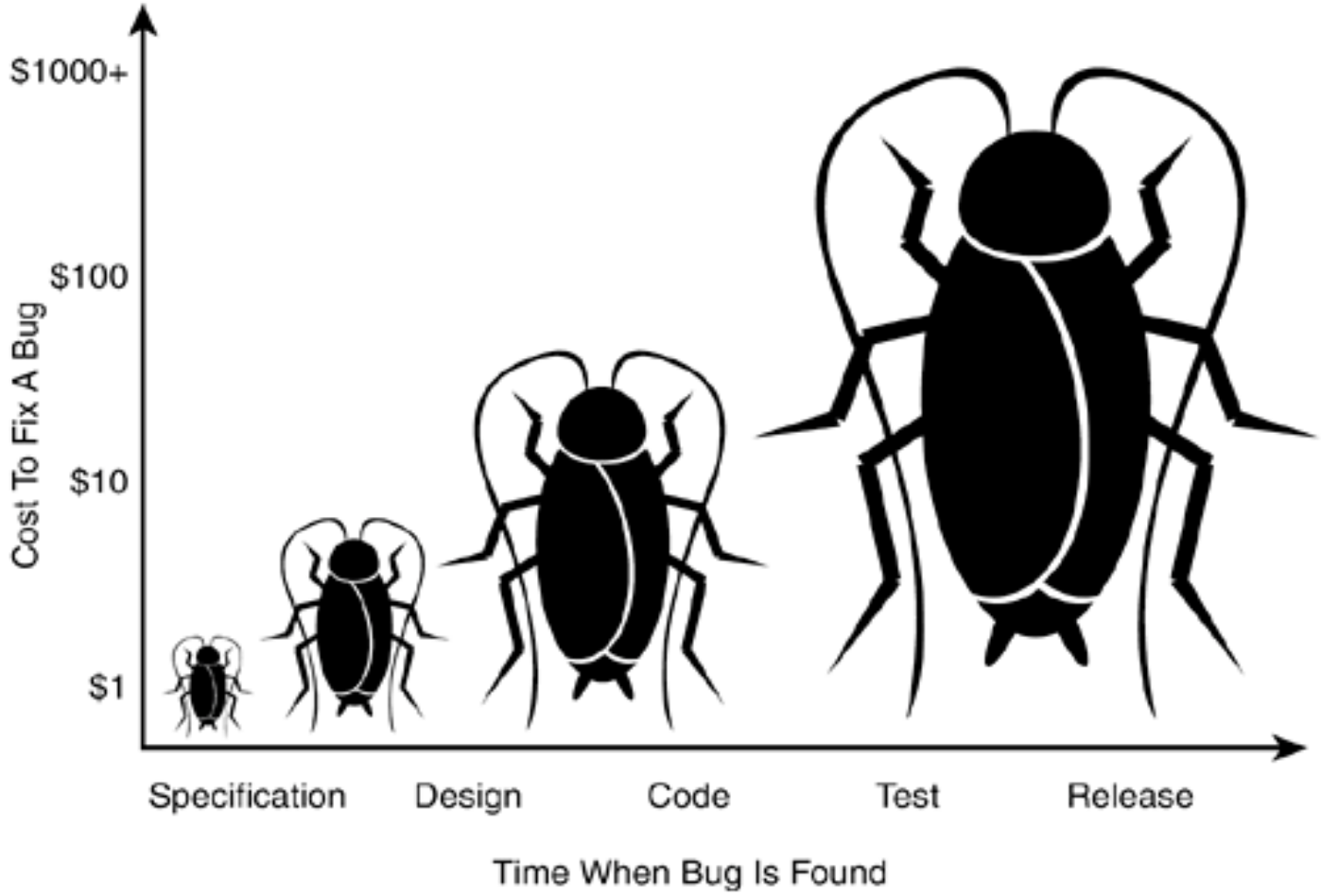
The Year 2000 (Y2K) Bug

- It's estimated that several hundred billion dollars were spent, worldwide, to replace or update computer programs.
- But this was not an end. On Jan. 1st, 2010, some taximeters in Wuhu and Liyang showed some weird information and did not work normally due to a novel Y2K Bug.
- So a deep research was undertaken. The research suggests that there may engender a new Y2K bug in Feb., 2038.

Why Do Defects Occur?



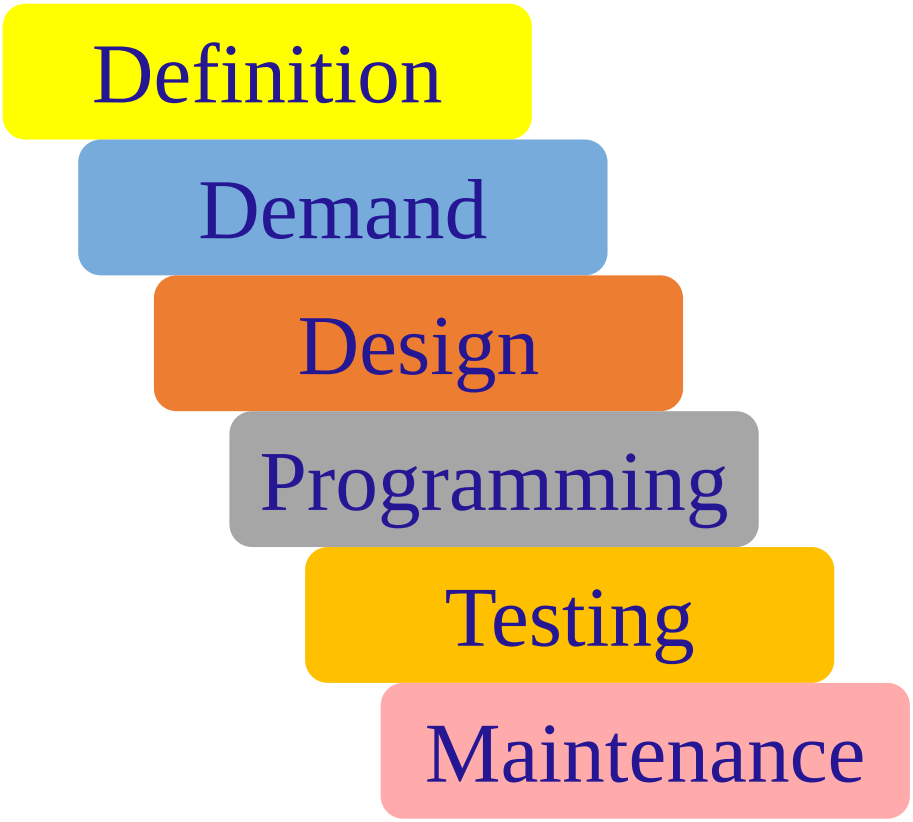
Cost of Defects



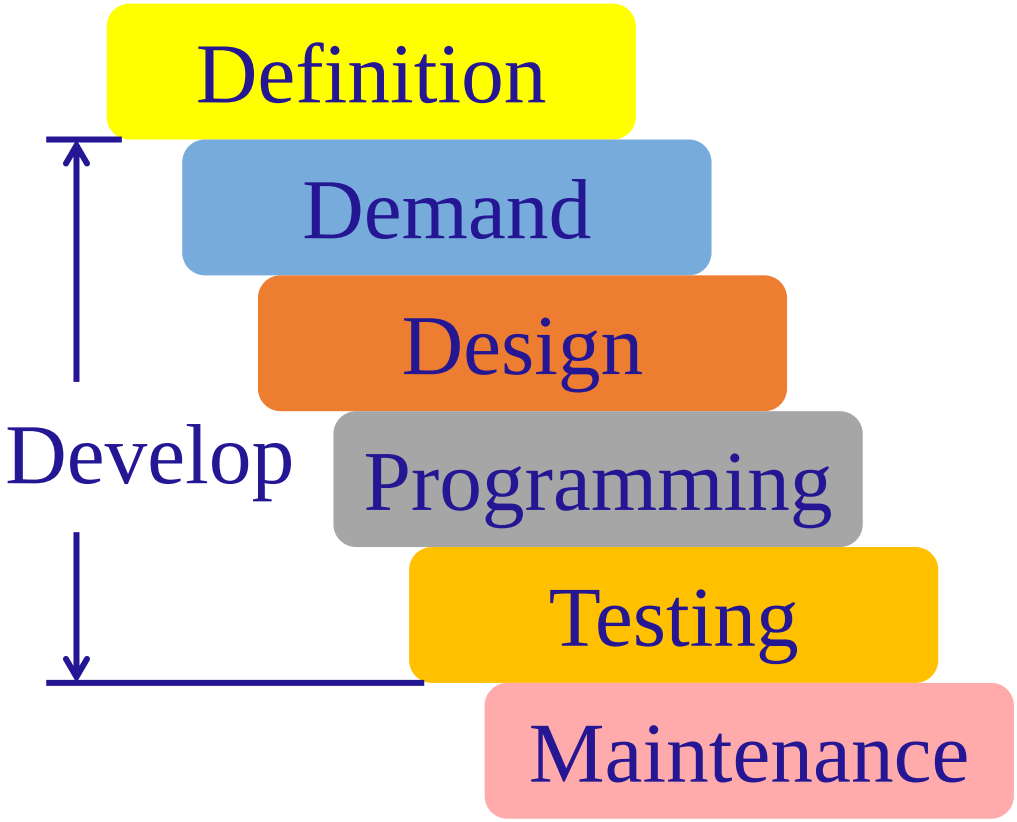
Software Testing

- 软件测试技术是对软件产品进行验证和确认的活动过程，其目的是尽快尽早发现软件产品中存在的诸问题，包括错误、缺陷以及用户预先定义需求的不一致性等。
- IEEE在1983年对软件测试定义如下：使用人工或者自动手段来运行或测定整个系统的过程，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。

Software Life Cycle



Software Life Cycle



Software Life Cycle

Definition

Demand

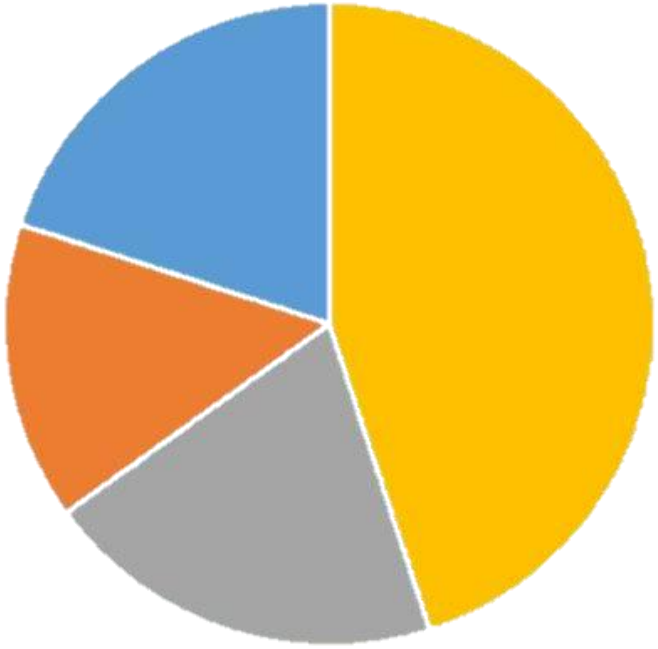
Design

Develop

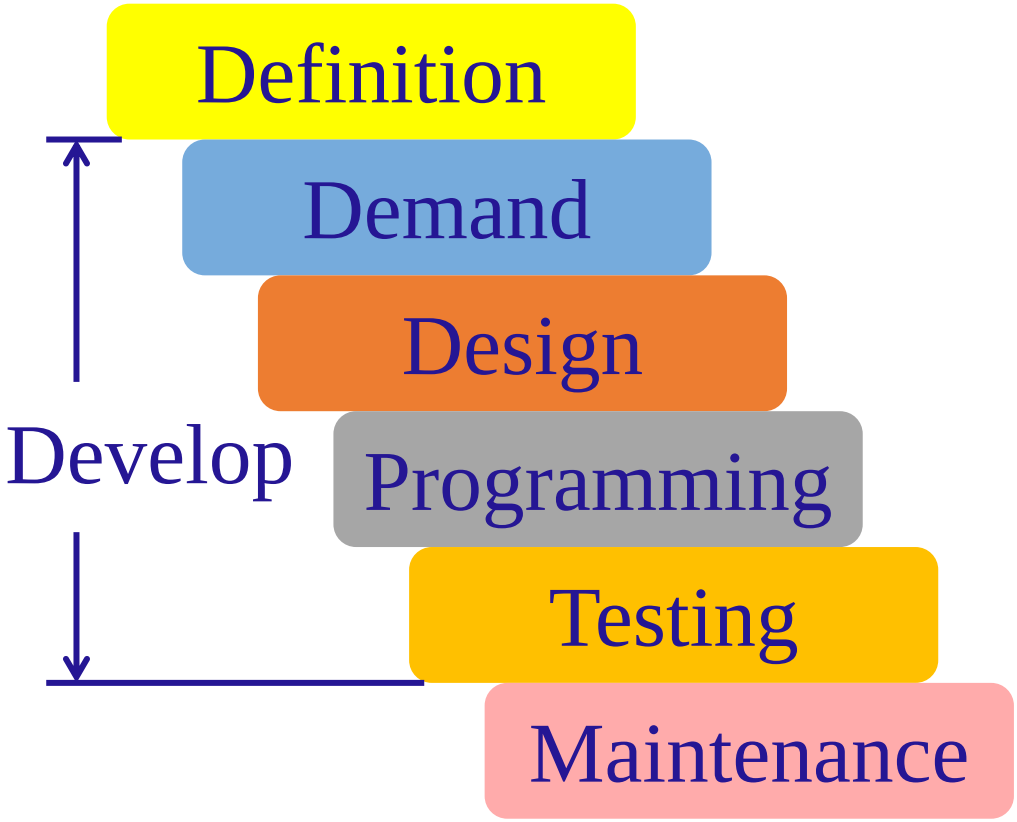
Programming

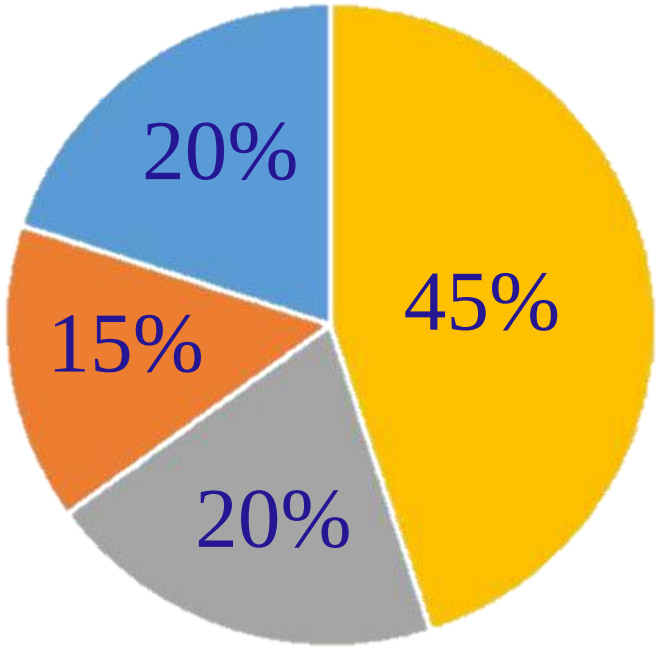
Testing

Maintenance

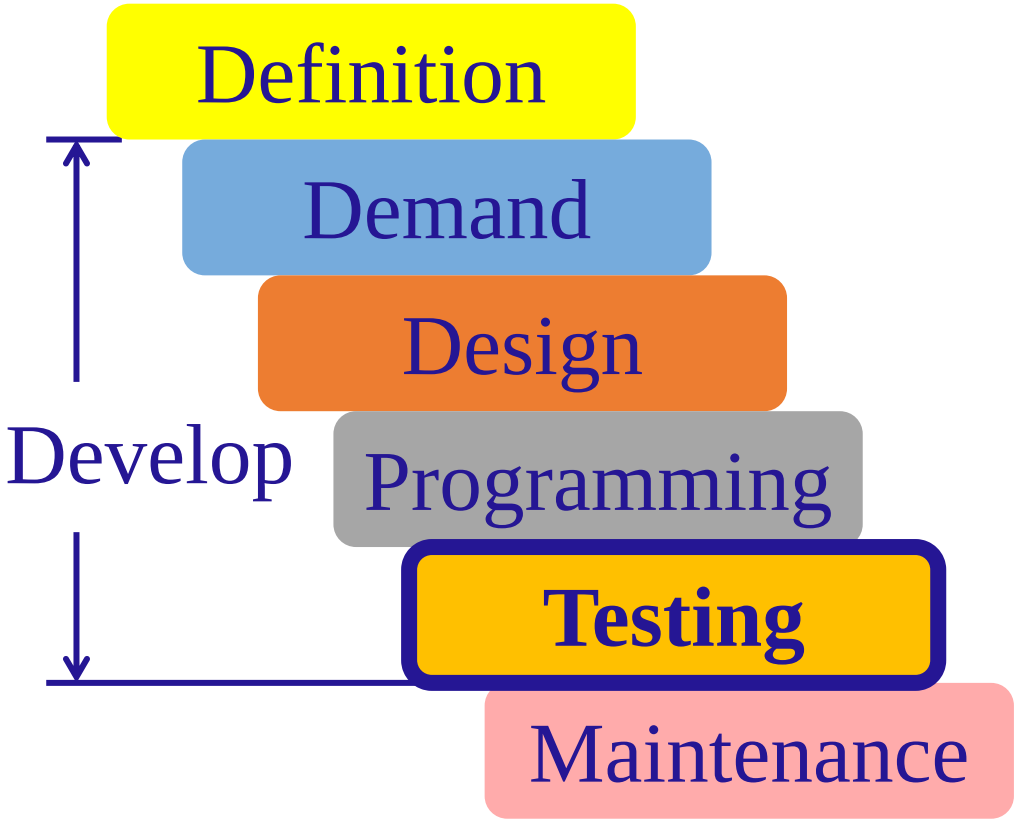


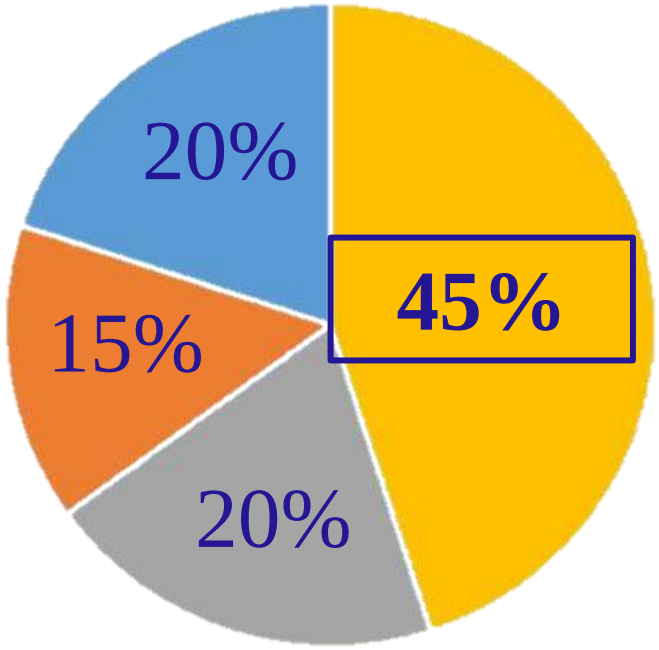
Software Life Cycle





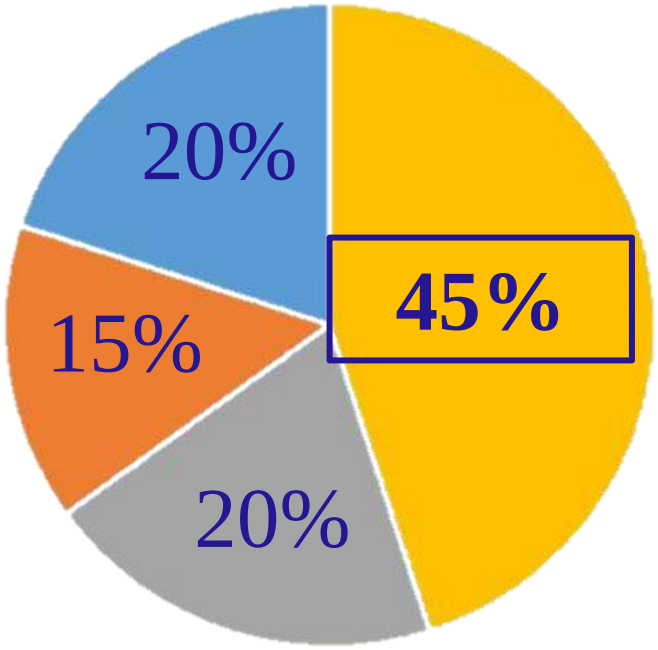
Software Life Cycle





Significance of Testing

- Software testing is of great significance, and it occupies almost half of the cycle.
- Were there no testing in the cycle, there would cause awful influence when we use software and thus tremendous cost would be consumed.



What Does a Software Tester Do?

- The goal of a software tester is to find bugs.
- The goal of a software tester is to find bugs and find them as early as possible.
- The goal of a software tester is to find bugs, find them as early as possible, and make sure they get fixed.

What Makes a Good Software Tester?

- They are explorers.
- They are troubleshooters.
- They are relentless.
- They are creative.
- They are pursuing perfection.
- They are tactful and diplomatic.
- They are persuasive.
- In addition to these traits, having some education in software programming is a big plus.

An Example of Software Testing

- 2018年1月，某同学在老师指导下使用合适的编程工具开发了一个能实现十进制数和IEEE754格式32位浮点十六进制数互转的软件。

- 此软件要求用户在界面上输入一个十进制数（分数或小数），能输出一个IEEE754格式32位以十六进制表示的浮点数，并能够给出对应符号、阶码和尾数三者细节信息。反之亦能。

An Example of Software Testing

- 此程序算法已经有现成的构思，因此只需要编程实现即可。该同学编写此程序代码花费了一天的时间，并做出了界面。该同学随即将软件交付测试人员。

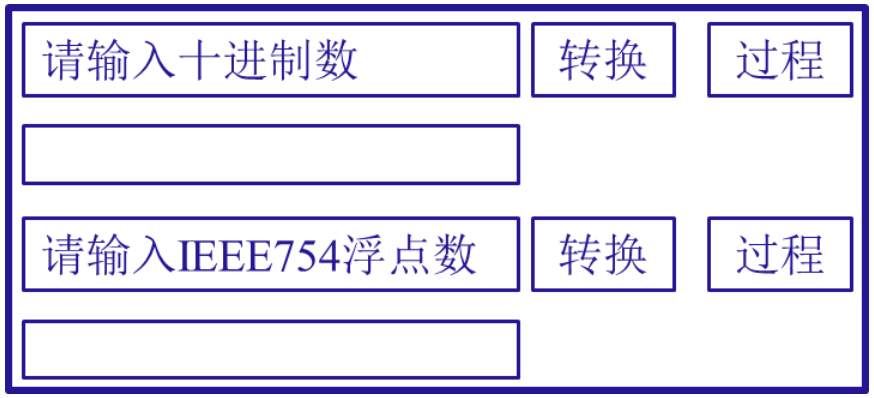
An Example of Software Testing

- 软件测试时，首先发现该软件缺少两个功能：

① 十进制输入只能是小数，不识别分数；

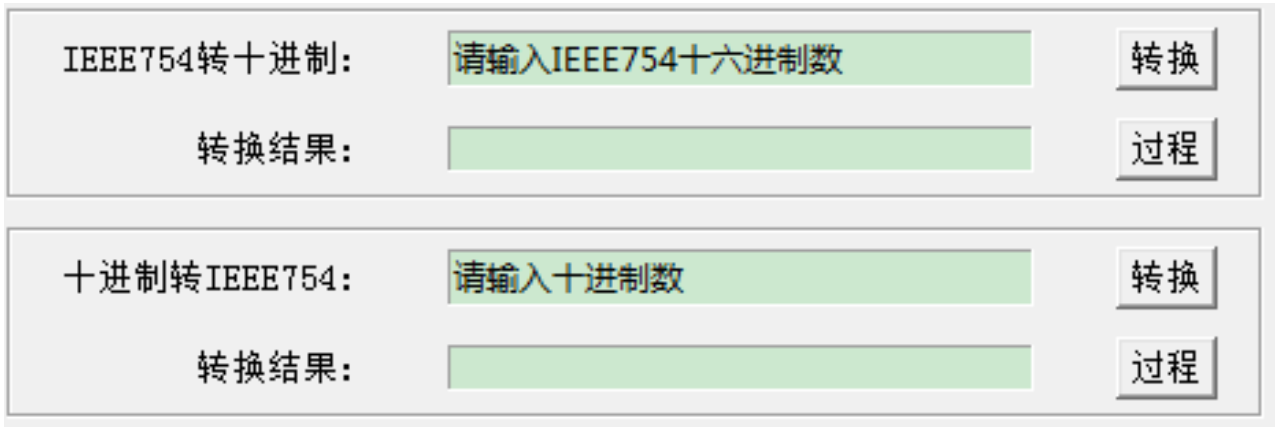
② 没有提供*S*、*E*和*M*三者细节信息。

- 因此，该软件从这一点来看和用户预先定义需求不一致。



An Example of Software Testing

- 该同学在返工后，将上述两个功能完善，达到了用户预先定义需求。于是测试人员对其功能进行进一步测试。



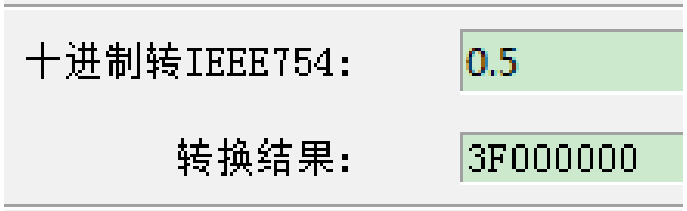
An Example of Software Testing

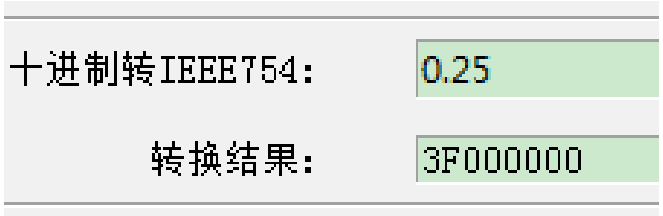
- 进一步测试发现了两个问题：

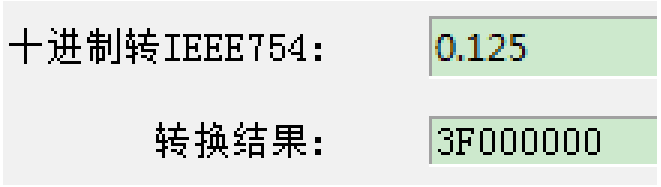
① 十进制数输入为0时，软件不报错；

② 十进制数输入某些不同小数时，输出结果却相同。

- 这说明该软件存在缺陷和错误，应找到问题根源并加以改正。

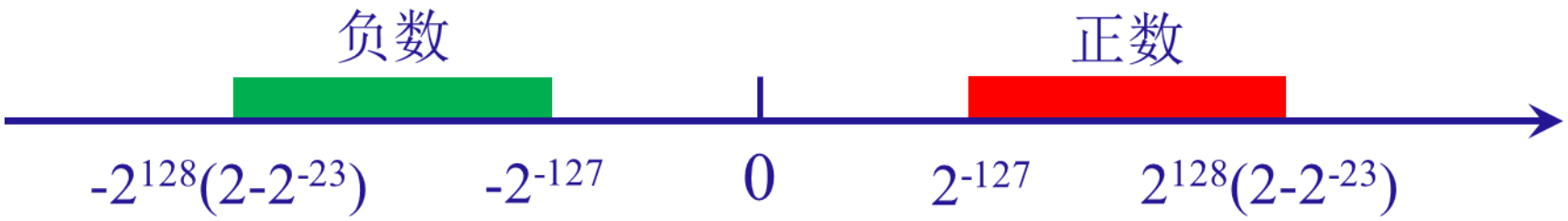






An Example of Software Testing

- 经过对代码进行断点调试，发现其中小数的数制转化代码出了问题，并且程序中也没有对输入十进制数的范围加以限定。该同学首先修正了小数的数值转化代码，然后在程序最前方加上了范围限定语句。



An Example of Software Testing

- 最后又经过了几次调试，最终基本实现了所需的基本功能，并且在所能覆盖的测试范围内转换无误。

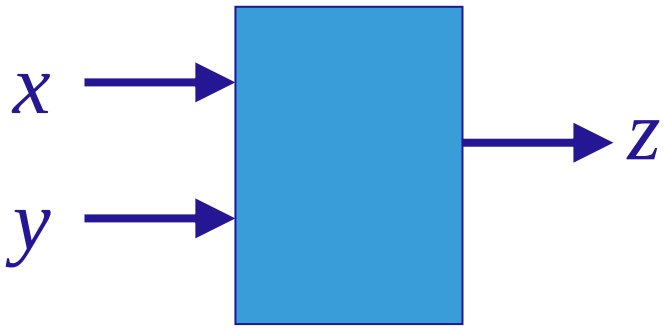
Aim of Software Testing

- Software testing is actually a process or a series of processes, identifying that the computer has achieved the functions it should accomplish and has not achieved the ones it did not mention. So the aim of software testing is **to find the defects**.
- One can never think that the aim can be defined as the certification of software validity.

Aim of Software Testing

Eg.1: Suppose a function $z=z(x,y)$, x and y are two integers. By using black-box testing on a 32-bit computer, how many test cases should be given?

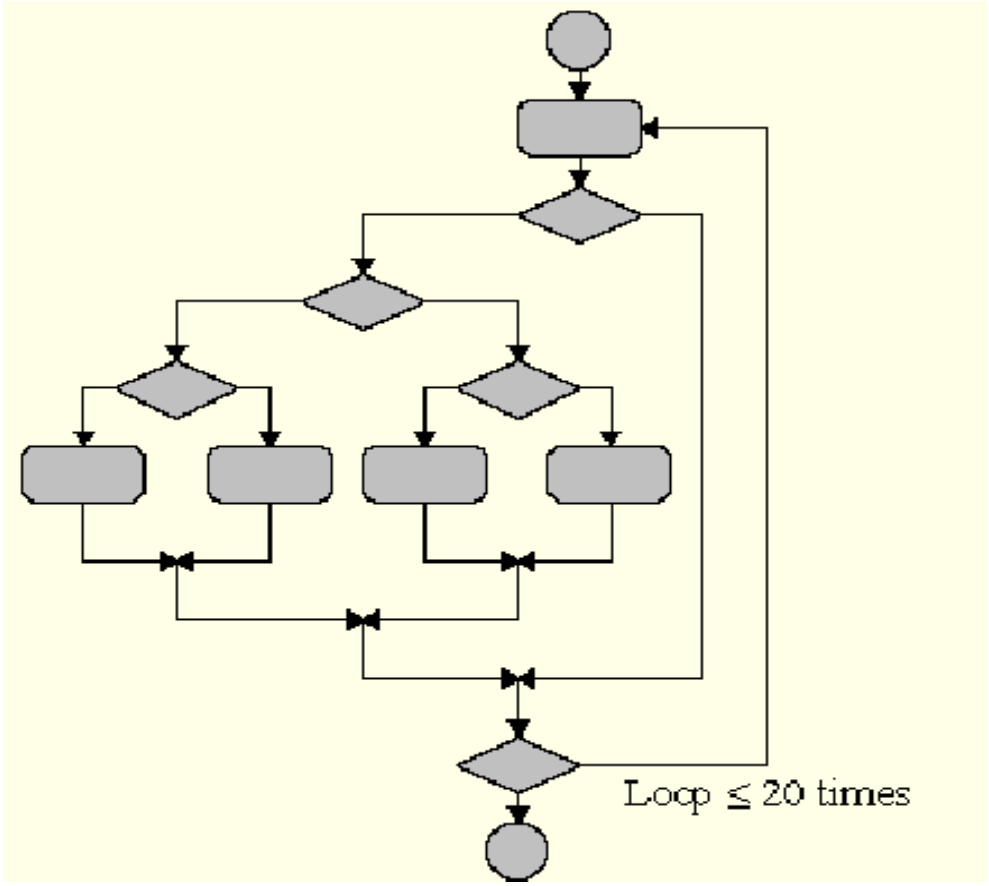
- 经计算，不同的测试数据组合数目是$2^{32}\times2^{32}=2^{64}>10^{19}$，以每秒执行1000个测试用例的速度计算，完成测试大概需要工作5亿年。



Aim of Software Testing

Eg.2: By using white-box testing, how many path covers can be found?

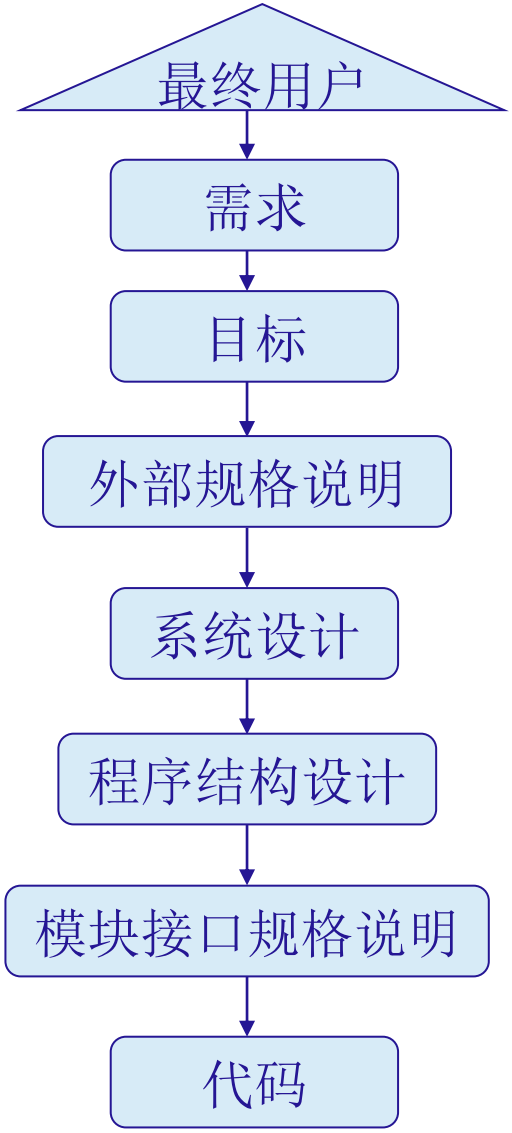
- 使用等比数列求和，可执行路径数大于10^{14}。若每秒执行1000个测试用例，完成所有可能路径的测试大概需要3170年。

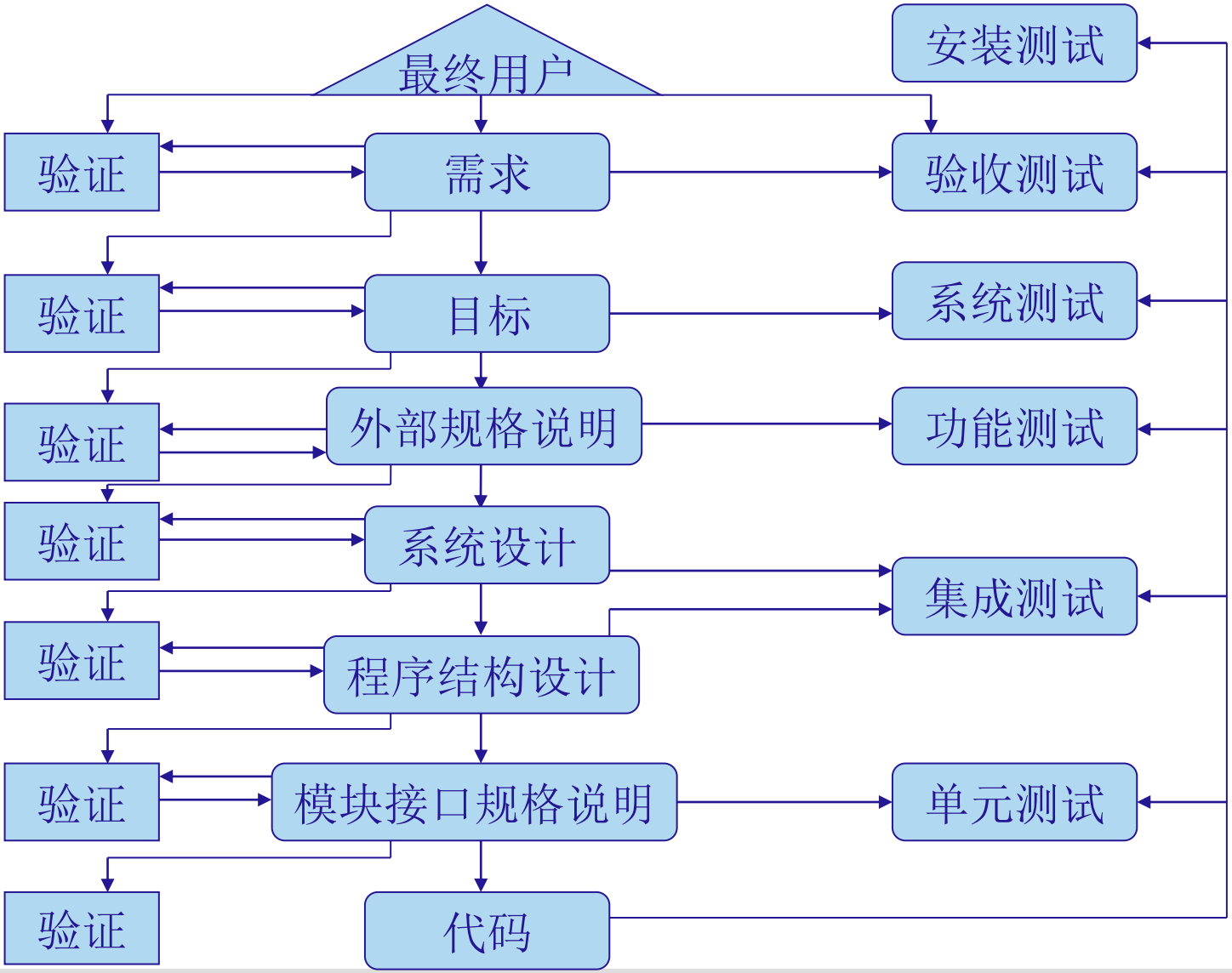


Techniques about Testing

- Static Black-Box Testing
- Dynamic Black-Box Testing
  - Equivalence Partitioning, Boundary Value Analysis
  - Decision Table, Cause-Effect Diagram
- Static White-Box Testing
- Dynamic White-Box Testing
  - Statement Coverage, Decision Coverage, Condition Coverage, Decision / Condition Coverage, Multiple Condition Coverage
  - Path Coverage

Development of Software




最终用户
安装测试
验证
需求
验收测试
验证
目标
系统测试
验证
外部规格说明
功能测试
验证
系统设计
集成测试
验证
程序结构设计
验证
模块接口规格说明
单元测试
验证
代码

THANK YOU!