

专业课

计算机

软件工程

袁礼

华图网校

版权所有 盗版必究

目录

软件工程.....	1
（一） 软件工程概述.....	1
1 软件危机.....	1
2 软件工程的定义.....	1
3 软件生命周期.....	2
（二） 需求分析.....	5
1 什么是需求分析.....	5
2 需求分析阶段的具体任务.....	6
3 功能需求和性能需求.....	7
4 需求分析中的问题解决.....	7
（三） 结构化分析与设计.....	8
1. 结构化分析概述.....	8
2 数据流图.....	9
3 数据字典.....	10
4 结构化设计概述.....	11
（四） 软件开发模型.....	13
1 软件开发模型.....	13
2 瀑布模型.....	13
3 其他软件开发模型.....	14
（五） 软件质量与项目管理.....	17
1 软件度量.....	17
2 软件质量因素.....	18
3 软件项目管理.....	18
4 软件质量保证的方法.....	19
（六） 软件测试和维护.....	20
1、软件测试概述.....	20
*软件测试的定义：.....	20
2、软件测试的概念原则.....	21
3 软件测试的方法和步骤.....	21
4 软件维护的定义.....	23

软件工程

(一) 软件工程概述

*对软件的思考

软件是一种逻辑实体，而不是具体的物理实体，具有抽象性。

软件的生产与硬件不同，在它的开发过程中没有明显的制造过程。

软件的运行和使用期间，没有硬件那样的机械磨损，老化问题。

软件的开发和运行常受到计算机系统的限制，对计算机系统有不同程度的依赖性。软件的开发复杂性高，成本昂贵。

软件的开发至今尚未完全摆脱手工艺的开发方式

*软件神话

有了软件开发书籍，就可以顺利完成软件开发了？

增加程序员就可以解决进度滞后？

只要程序代码质量高，项目就成功，文档不是必要的？

软件质量只能在正式投入运行后才知道？

1 软件危机

软件开发的生产率远远不能满足客观需要。开发的软件产品往往不能满足用户的实际需要。软件产品的质量低下且可维护性差。

很难估计软件开发的进度计划与成本。

软件开发是一项系统工程

2 软件工程的定义

Boehm: 运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料

IEEE: 软件工程是开发、运行、维护和修复软件的系统方法

Fritz Bauer: 建立并使用完善的工程化原则，以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法

*软件工程三要素：方法、工具和过程

软件工程方法为软件开发提供了“如何做”(How)的方法和步骤。

软件工具为软件工程方法提供了自动的或半自动的软件支撑环境。

*软件工程过程定义了：

1. 方法使用的顺序
2. 要求交付的文档资料
3. 为保证质量和适应变化所需要的管理
4. 软件开发各个阶段完成的里程碑

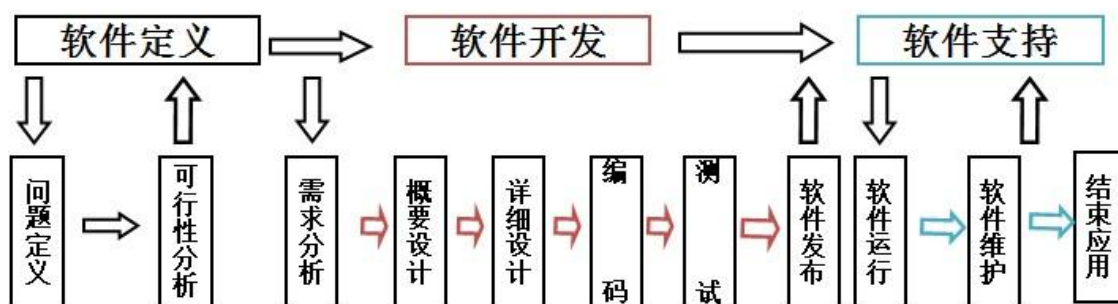
***软件工程项目的基本目标**

1. 付出较低的开发成本
2. 达到要求的软件功能
3. 取得较好的软件性能
4. 开发的软件易于移植
5. 需要较低的维护费用
6. 能按时完成开发工作，及时交付使用

***软件工程的各个阶段**

软件工程的各个阶段		
软件定义阶段	软件开发阶段	软件维护阶段
<ul style="list-style-type: none"> • 问题定义 • 可行性分析 • 立项或签订合同 	<ul style="list-style-type: none"> • 需求分析 • 软件设计 • 编码与测试 • 发布或安装验收 	<ul style="list-style-type: none"> • 软件的使用、运行 • 软件维护

3 软件生命周期



***概要设计**

设计系统结构

将复杂系统划分为模块

确定每个模块的功能

确定模块之间的调用关系

确定模块之间的接口，即传递的信息

评价模块结构的质量

设计数据结构

编写概要设计文档

评审概要设计文档

***详细设计**

为每一个模块确定算法和局部数据结构，用某种选定的表达工具表示算法和数据结构的细节。

常见过程设计工具：

图形工具：程序流程图，NS 图，PAD 图,HIPD 图

表格工具：判定表

语言国内国际：PDL（伪码）

***软件设计的基本原理**

抽象化

模块化

信息隐蔽

模块独立

内聚性和耦合性是评判软件模块独立性的两个标准。

高内聚，低耦合。

***内聚性**

模块内各个元素彼此结合的紧密程度。

以下为内聚性由弱到强：

偶然内聚：无关联

逻辑内聚：由参数确定逻辑关系

时间内聚：需要同时或顺序执行，如定义变量，赋初值

过程内聚：必须以特定顺序执行

通信内聚：使用公共数据

顺序内聚：需要前一个处理单元的输出作为后续单元输入

功能内聚：共同完成功能，不可再分

*耦合性

模块间相连接的紧密程度。

以下为耦合性由高到低：

内容耦合：一个模块直接访问另一个

公共耦合：一组模块同时访问一个全局数据结构

外部耦合：一组模块都访问同一全局简单变量

控制耦合：明显把变量信息送入另一模块

标记耦合：通过记录传递数据

数据耦合：两个模块通过数据参数交换信息

非直接耦合：没有直接关系

【练习】内聚性和耦合性是度量软件模块独立性的重要准则，软件设计时应力求（B）。

- A. 高内聚，高耦合 B. 高内聚，低耦合
- C. 低内聚，高耦合 D. 低内聚，低耦合

【练习】在详细设计阶段，经常采用的工具有（A）

- A. PAD
- B. SA
- C. SC
- D. DFD

【练习】软件生存周期包括以下哪几个阶段（ACE）

- A、软件定义
- B、软件使用
- C、软件开发
- D、软件测试
- E、软件维护

【练习】软件危机的主要表现有多个方面，包括（ABEF）

- A. 需求难以满足
- B. 生产成本难以控制
- C. 进度无法控制

- D. 需求定义不准确
- E. 质量不易保证
- F. 可维护性差

【练习】在以下几种耦合中，最低程度的耦合是（C）。

- A、数据耦合
- B、内容耦合
- C、非直接耦合
- D、公共耦合

（二）需求分析

1 什么是需求分析

*重要性

需求分析是发现、求精、建模、规格说明和复审的过程；

需求分析是系统设计的基础，关系到程序的成败和软件产品的质量。

*需求获取困难,原因有三:

- 一是用户需求的动态性(不稳定性)
- 二是需求的模糊性(不准确性)
- 三是需求必须得到用户的确认，否则毫无意义

*需求分析的任务

需求分析的基本任务是准确地回答“系统必须做什么?”这个问题。目标系统提出完整、准确、清晰、具体的要求。

1. 确定对系统的综合要求
2. 分析系统的数据要求
3. 导出系统的逻辑模型
4. 修正系统开发计划
5. 开发原型系统
6. 需求分析的任务

2 需求分析阶段的具体任务

一、确定对系统的综合要求

1. 系统功能要求

应该划分出系统必须完成的所有功能。

2. 系统性能要求

例如，联机系统的响应时间，系统需要的存储容量以及后援存储，重新启动和安全性等方面

3. 运行要求

这类要求集中表现为对系统运行时所处环境的要求。例如，支持系统运行的系统软件是什么，采用哪种数据库管理系统，需要什么样的外存储器和数据通信接口等。

4. 将来可能提出的要求

应该明确地列出那些虽然不属于当前系统开发范畴，但是据分析将来很可能会提出来的要求。这样做的目的是在设计过程中对系统将来可能的扩充和修改预做准备，以便一旦需要时能比较容易地进行这种扩充和修改。

二、分析系统的数据要求

任何一个软件系统本质上都是信息处理系统，系统必须处理的信息和系统应该产生的信息在很大程度上决定了系统的面貌，对软件设计有深远影响，因此，必须分析系统的数据要求，这是软件需求分析的一个重要任务。分析系统的数据要求通常采用建立概念模型的方法。

三、导出系统的逻辑模型

综合上述两项分析的结果可以导出系统的详细的逻辑模型，通常用数据流图、数据字典和主要的处理算法描述这个逻辑模型。

四、修正系统开发计划

根据在分析过程中获得的对系统的更深入更具体的了解，可以比较准确地估计系统的成本和进度，修正以前制定的开发计划。

五、开发原型系统

开发原型系统通常有两个主要目的：检验关键设计方案的正确性及系统是否真正满足用户的需要。

原型系统使用户通过实践获得关于未来的系统将怎样为他们工作的更直接更具体的概念，从而可以更准确地提出和确定他们的要求。

*需求分析四阶段

需求获取

需求分析

编写需求规格说明书

需求评审

3 功能需求和性能需求

用户需求通常可以分为功能需求和性能需求两类。

功能需求定义了系统应该做什么，系统要求输入什么信息，输出什么信息，以及如何将输入变换为输出。

性能需求则定义了软件运行的状态特征，如系统运行效率，可靠性，安全性，可维护性等等。

*需求分析应获取的内容

应该获取用户需求的内容包括：

(1)物理环境。系统运行的设备地点、位置是集中式的还是分布式的，对环境的要求如何(如温度、湿度，电磁场干扰等)。

(2)系统界面。要求与其他系统进行数据交换的内容与格式，终端用户的类型与熟练程度，用户对界面的特定要求，用户操作的易接受性等。

(3)系统功能。系统应该完成的功能以及何时完成，对于系统运行速度、响应时间或者数据吞吐量的要求，系统运行的权限规定，系统可靠性要求，是否要求可移植，未来扩充或者升级的要求。

(4)数据要求。输入输出数据的种类与格式，计算必须达到的精度，数据接收与发送的频率，数据存储的容量和可靠性，数据或者文件访问的控制权限，数据备份的要求。

(5)系统文档规格。系统要求交付什么文档，各类文档的编制规范和预期使用对象。

(6)系统维护要求。系统出错后可以允许的最大恢复时间，对错误修改的回归测试要求，系统运行日志规格，是否允许对系统修改，系统变化如何反映到设计中。

4 需求分析中的问题解决

在获取需求过程中遇到的典型问题及其解决方法是：

(1)如何理解问题。大多数情况下，软件开发人员不是问题领域的行家。但是要准确、完整的获取需求必须对问题具有深入的理解与把握。许多问题即使是用户业务人员也可能没有自觉的认识。

(2)分析员与用户的沟通问题。分析员对问题的理解

必须从信息处理要求出发，而用户更多的考虑是本身的业务领域。与用户建立相互信任、有效的沟通是分析员的首要任务。

(3)用户需求的可变性。

用户需求通常是不断变化的，而软件开发人员则希望将需求冻结在某一时刻。影响用户需求变化的因素可以是用户领域的业务扩充或者转移，市场竞争的要求，用户主管人员的变更等。现实情况是分析员只能接受需求不断变化的事实，应该千方百计地使其工作适应需求的变化。

现实世界是复杂多变的。为了将现实世界中问题的求解映射为信息处理模型，对问题进行分解与抽象是普遍有效的基本法则。

*需求规格说明书

作用：

便于用户、开发人员进行理解和交流

反映出用户问题的结构

作为确认测试和验收的依据

特点：

正确性、无歧义、完整性、可验证性

一致性、可理解性、可修改性、可跟踪性

【练习】软件需求分析阶段的工作,可以分为 4 个方面:需求获取,需求分析,编写需求规格说明书以及(B)

- A.用户
- B.需求审评
- C.总结
- D.都不正确

【练习】需求分析中开发人员要从用户那里了解 (A)

- A. 软件做什么
- B. 用户使用界面
- C. 输入的信息
- D. 软件的规模

【练习】原型化方法是用户和设计者之间执行的一种交互构成，适用于 (A) 系统。

- A. 需求不确定性高的
- B. 需求确定的
- C. 管理信息
- D. 实时

(三) 结构化分析与设计

1. 结构化分析概述

- 结构化方法

- 结构化分析思想
- 结构化分析模型

*结构化方法

一种面向数据流的传统软件开发方法

以数据流为中心构建软件的分析模型和设计模型

分为:

结构化分析(Structured Analysis 简称 SA)

结构化设计(Structured Design 简称 SD)

结构化程序设计(Structured Programming 简称 SP)

*结构化分析思想

抽象:从作为整体的软件系统开始(第一层),每一抽象层次上只关注于系统的输入输出

分解:自顶向下逐层分解,不断分解为子系统,模块……

随着分解层次的增加,抽象的级别越来越低,也越接近问题的解(算法和数据结构)

*结构化分析模型

DD (数据描述): 核心, 包含所有数据的描述

DFD (数据流图): 用于功能建模, 描述系统的输入数据流如何经过一系列的加工变换逐步变换成系统的输出数据流

ERD (实体关系图): 用于数据建模, 描述 DD 中数据之间的关系

状态转换图: 用于行为建模, 描述系统接收哪些外部事件, 以及在外部事件的作用下的状态迁移情况

2 数据流图

Data Flow Diagram(简称 DFD): 描述输入数据流到输出数据流的变换(即加工)过程, 用于对系统的功能建模。

DFD基本图形符号

- 数据流(data flow): 由一组固定成分的数据组成, 代表数据的流动方向
- 加工(process): 描述了输入数据流到输出数据流的变换, 即将输入数据流加工成输出数据流
- ═ 存储文件(file): 使用文件、数据库等保存某些数据结果供以后使用
- 源或潭(source or sink): 存在于系统之外的人员或组织, 代表输入数据的来源或输出数据的去向

3 数据字典

*数据字典条目的组成

- 数据流图与数据字典是密不可分的, 两者结合起来构成软件的逻辑模型(分析模型)
- 数据字典由字典条目组成, 每个条目描述 DFD 中的一个元素
- 数据字典条目包括: 数据流、文件、数据项(组成数据流和文件的数据)、加工、源或宿

*数据字典的描述符号

符 号	意 义	举 例
=	定义为	x=... 表示x由...组成
+	与	a+b 表示a和b
[... ...]	或	[a b] 表示a或b
n { } m	重复	用户名= 6 { 字母} 10
(...)	可选	(营业员) 表示营业员可选
**	注释	
..	连接符	月= "01" .. "12"

***数据流组成示例(发票)**

发票=单位名称+{商品名+数量+单价+金额}
+总金额+日期+(营业员)

单位名称

商品名	数量	单价	金额
总金额			

日期

营业员

4 结构化设计概述

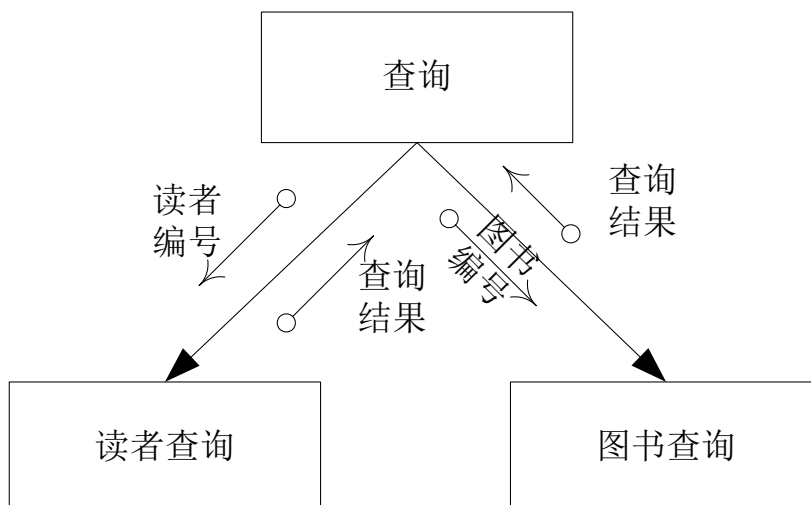
结构化设计(Structured Design, 简称 SD)是将结构化分析得到的数据流图映射成软件体系结构的一种设计方法。强调模块化、逐步求精、信息隐蔽、高内聚低耦合等设计准则。分为概要设计和详细设计两大步骤

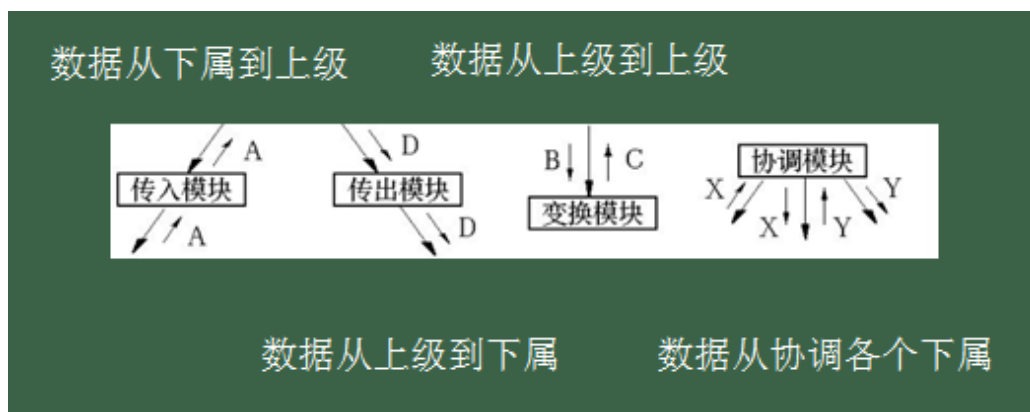
*结构图(Structure Chart)

描述一个软件系统的模块划分与层次，表示模块之间的调用关系、模块间信息流与控制流的传递关系及模块与外界或数据存储的信息接口。

结构图的基本成分有：模块、调用和数据

结构图一般不列入设计文档，只用于设计阶段检查模块设计的正确性和模块独立性。





*结构化设计的步骤

- 1) DFD 映射到初始结构图
- 2) 根据设计准则和启发式策略改进结构图
- 3) 书写设计文档,为每个模块书写模块的功能、接口、约束和限制等
- 4) 设计评审

【练习】结构化分析方法就是面向(B)的自顶向下逐步求精进行需求分析的方法。

- A).目标
- B).数据流
- C).功能
- D).对象

【练习】数据流图用于抽象描述一个软件的逻辑模型，下列图符名属于数据流图合法标示的是（ BCD ）

- A、控制流
- B、加工
- C、数据存储
- D、源和潭

【练习】数据流图和(C)共同组成系统的逻辑模型。

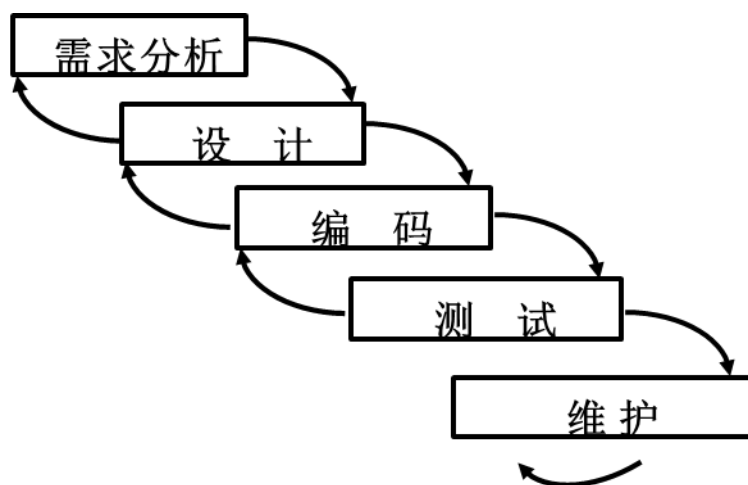
- A).HIPO 图
- B).PDL
- C).数据字典
- D).层次图

（四）软件开发模型

1 软件开发模型

软件开发模型(Software Development Model)是指软件开发全部过程、活动和任务的结构框架。软件开发包括需求、设计、编码、测试、维护阶段。软件开发模型能清晰、直观地表达软件开发全过程,明确规定了要完成的主要活动和任务,用来作为软件项目工作的基础。

2 瀑布模型



***瀑布模型的特点和问题:**

1. 阶段间具有顺序性和依赖性
2. 每个阶段必须完成规定的文档;
3. 每个阶段结束前完成文档审查, 尽早改正错误,

***存在的问题:**

1. 开发过程一般不能逆转, 否则代价太大。
2. 实际的项目开发很难严格按该模型进行。
3. 客户往往很难清楚地给出所有的需求, 而该模型却要求如此。
4. 软件的实际情况必须到项目开发的后期才能看到, 不利于及时纠正问题。

***瀑布模型: 需求分析**

1. 确定要开发软件系统的总目标
2. 对用户提出的要求进行分析并给出详细的定义

3. 完成可行性分析
4. 编写软件需求说明书或系统功能说明书及初步的系统用户手册
5. 提交管理机构评审

***瀑布模型：软件设计**

概要设计 — 把各项需求转换成软件的体系结构。结构中每一组成部分都是意义明确的模块，每个模块都和某些需求相对应

详细设计 — 对每个模块要完成的工作进行具体的描述，为源程序编写打下基础

编写设计说明书，提交评审。

***瀑布模型：程序编写**

把软件设计转换成计算机可以接受的程序代码，即写成以某一种特定程序设计语言表示的“源程序清单”

写出的程序应当是结构良好、清晰易读的，且与设计相一致的

内聚性和耦合性

***瀑布模型：软件测试**

单元测试，查找各模块在功能和结构上存在的问题并加以纠正

组装测试，将已测试过的模块按一定顺序组装起来

按规定的各项需求，逐项进行有效性测试，决定已开发的软件是否合格，能否交付用户使用

***瀑布模型：运行 / 维护**

改正性维护 运行中发现了软件中的错误需要修正

适应性维护 为了适应变化了的软件工作环境，需做适当变更

完善性维护 为了增强软件的功能需做变更

【练习】瀑布模型的存在问题是（ B ）

- A 用户容易参与开发
- B 缺乏灵活性
- C 用户与开发者已沟通
- D 适用可变需求

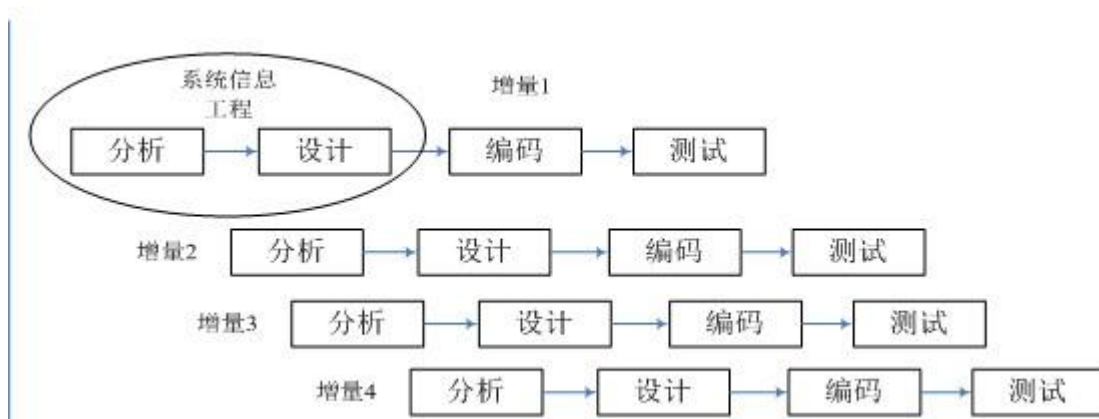
3 其他软件开发模型

- 原型模型
- 增量模型
- 螺旋模型
- 喷泉模型

***原型模型**



***增量模型**

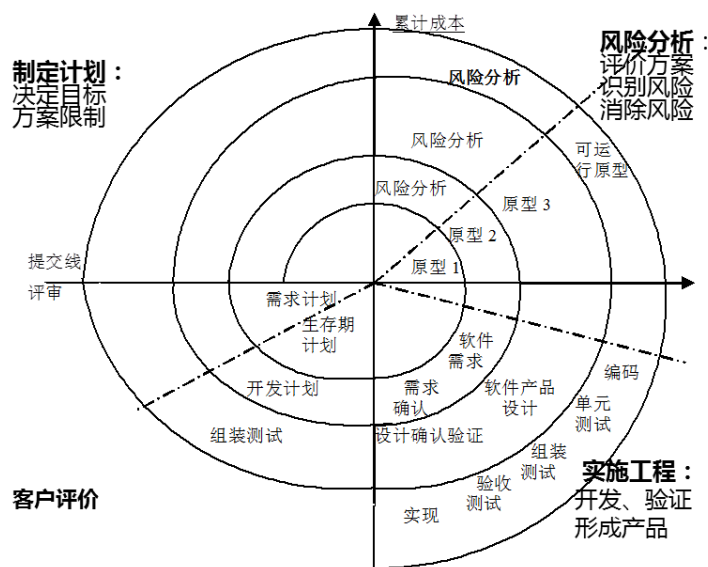


不断增加，逐渐形成

增量模型融合了线性顺序模型的基本成分(重复地应用)和原型的迭代特征。增量模型采用随着日程时间的进展而交错的线性序列。每一个线性序列产生软件的一个可发布的“增量”。

增量模型是一种非整体开发的模型。软件在该模型中是“逐渐”开发出来的。该模型有较大的灵活性，适合于软件需求不明确、设计方案有一定风险的软件项目。

***螺旋模型**



这种模型适合于大型系统的开发，应该说它对于具有高度高风险的大型复杂软件系统的开发是最为实际有效的方法。

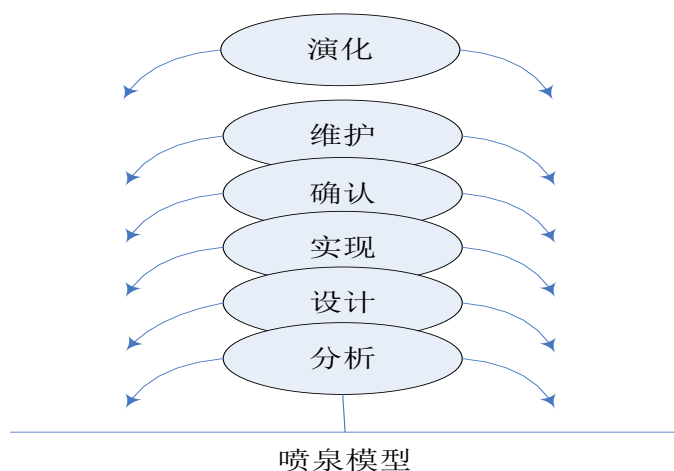
螺旋模型沿着螺线旋转，在笛卡尔坐标的四个象限上分别表达了四个方面的活动：

- (1) 制定计划——确定软件目标，选定实施方案，弄清项目开发的限制条件；
- (2) 风险分析——分析所选方案，考虑如何识别风险和消除风险；
- (3) 实施工程——实施软件开发；
- (4) 客户评估——评价开发工作，提出修正建议。

沿螺线自内向外每旋转一圈便开发出更为完善的一个新的软件版本。

*喷泉模型

喷泉模型体现了软件创建所固有的迭代和无间隙的特征。系统某个部分常常重复工作多次，相关功能在每次迭代中不断改进。



【练习】在瀑布模型中，将软件划分为若干个阶段，软件项目的可行性研究一般归属于（B）

- A. 维护阶段
- B. 需求阶段
- C. 运行阶段
- D. 开发阶段

【练习】螺旋模型是一种演进式的软件过程模型，结合了原型开发方法的系统性和瀑布模型的可控性的特点。它有个显著特点，采用（C）的方式逐步加深系统定义和实现的深度，降低风险。

- A. 逐步交付
- B. 顺序
- C. 循环
- D. 增量

【练习】喷泉模型描述了(1)的开发模型，它体现了这种开发方法创建软件的过程所固有的(2)和(3)的特征。

A (1)

- A、 面向对象 B、 面向数据流
- C、 面向数据结构 D、 面向事件驱动

C (2)

- A、 归纳 B、 推理 C、 迭代 D、 递归

A (3)

- A、 开发各阶段之间无“间隙” B、 开发各阶段分界明显
- C、 部分开发阶段分界明显 D、 开发过程不分段

（五） 软件质量与项目管理

1 软件度量

软件度量（software measurement）是对软件开发项目、过程及其产品进行数据定义、收集以及分析的持续性定量化过程，目的在于对此加以理解、预测、评估、控制和改善。没有软件度量，就不能从软件开发的暗箱中跳将出来。通过软件度量可以改进软件开发过程，促进项目成功，开发高质量的软件产品。

*软件度量三维度

软件度量包括 3 个维度，即项目度量、产品度量和过程度量。

项目度量是针对软件开发项目的特定度量，目的在于度量项目规模、项目成本、项目进

度、顾客满意度等，辅助项目管理进行项目控制。

软件产品度量用于对软件产品进行评价，并在此基础上推进产品设计、产品制造和产品服务优化。软件产品的度量实质上是软件质量的度量

过程度量是对软件开发过程的各个方面进行度量，目的在于预测过程的未来性能，减少过程结果的偏差，对软件过程的行为进行目标管理，为过程控制、过程评价持续改善提供定量性基础。过程度量与软件开发流程密切相关，具有战略性意义。

2 软件质量因素

- (1)正确性
- (2)健壮性
- (3)效率
- (4)完整性(安全性)
- (5)可用性
- (6)易用性
- (7)风险
- (8)可理解性
- (9)可维护性
- (10)灵活性(适应性)
- (11)可测试性
- (12)可移植性
- (13)可再用性
- (14)互运行性

3 软件项目管理

为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对成本、人员、进度、质量、风险等进行分析和管理的活动。

***软件项目管理的四项任务：**

1. 软件项目计划与组织
2. 软件项目成本管理
3. 软件项目进度控制
4. 软件质量保证

***软件项目计划与组织**

工作内容：

- 1、深入了解项目基本情况
- 2、制定项目开发计划
- 3、建立合理的项目组织结构

***软件项目成本管理的内容：**

1. 资源计划：确定为完成项目各活动需什么资源和这些资源的数量。
2. 成本估算：对项目所需要的人力(以人月为单位)、项目持续时间(以年份或月份为单位)、项目成本(以元为单位)做出估算。
3. 成本预算：把估算的总成本分配到各个工作细目，建立基准成本用来衡量项目的成本执行情况。
4. 成本控制：及时发现成本偏差，并做出正确反应。

***软件项目进度控制**

进度控制就是比较项目执行的实际状态和项目计划之间的差异，并做出必要的调整使项目向有利的方向发展。



4 软件质量保证的方法

- 1、建立软件质量保证小组；
- 2、选择和确定软件质量保证活动；
- 3、制定和维护质量保证计划；
- 4、执行质量保证计划、对相关人员进行培训、选择与整个软件工程环境相适应的质量保证工具；
- 5、不断完善质量保证过程活动中存在的不足，改进项目的质量保证过程。

***软件质量保证体系与实施**

为了顺利开展质量保证活动，必须事先明确项目双方以及项目开发方部门间的质量保证责任及任务，并对质量保证活动进行合理评价及监控。这些内容构成了质量保证体系。

质量保证体系的实施：

- 1.明确双方职责
- 2.使用合理的质量评价指标体系
- 3.建立监控体系实施质量保证

【练习】软件质量保证是面向（c）的活动

- A 用户要求
- B 开发者
- C 消费者
- D 软件需求

【练习】进行软件质量管理的重要性有：（ABCD）

- A、维护降低成本
- B、法律上的要求
- C、市场竞争的要求
- D、质量标准化的趋势
- E、软件工程的需要

【练习】一般项目和大型项目管理的区别包括(ABCD)

- A、项目周期
- B、项目组织
- C、管理模式
- D、沟通协作

（六） 软件测试和维护

1、软件测试概述

***软件测试的定义：**

软件测试是为了发现缺陷而执行程序的过程。

测试的目的是想以最少的时间和人力找出软件中潜在的各种错误和缺陷。测试只能尽可能多的查找出程序中错误，而不能证明程序中没有错误。

***理解软件测试：**

- 1.测试是为了证明程序中有错误，而不是证明程序中无错误。
- 2.一个好的测试用例（Use Case）指的是它可能发现至今尚未发现的缺陷。
- 3.一次成功的测试指的是发现了新的软件缺陷的测试。

* **用例 (Use Case)**: 系统如何反应外界请求的描述, 是一种通过用户的使用场景来获取需求的技术。

*软件测试概述

1. 软件测试的范围并不只是对编码阶段的语法错、语义错、运行错进行查找的一系列活动。而是对软件计划、软件设计、软件编码进行查错和纠错的活动。

2. 它涉及到软件开发周期中各个阶段的错误, 并分析错误的性质与位置而加以纠正。纠正过程可能涉及到改正或重新设计相关的文档活动。找错的活动称软件测试, 纠错的活动称软件调试。

2、软件测试的概念原则

1. 错误(error)、缺陷(fault)和故障(failure)

人们在进行软件开发的过程中犯了一个错, 则称为一个错误(error)。应用到测试过程时, 有两种不同的使用方式。在第一种使用方式中, 错误是指一个实际测量值与理论预期值之间的差异, 这种差异就是错误; 第二种使用方式中, 错误是指一些人的行为引起的软件中的某种故障, 通常这些故障是由软件错误造成的。

缺陷(fault)常被称为 bug, 它是导致软件失败的一个条件。当开发人员犯了一个错, 就会在软件中引入一个或多个缺陷。

故障(failure)又称失效, 它是指软件不能按软件规格说明要求执行, 从而引起软件行为与用户需求的的不一致现象。失效可能发生在测试阶段, 也可能发生在软件交付之后的运行阶段和维护阶段。

缺陷是开发人员所看到的软件系统的内部问题, 而故障是用户从外部观察到的软件行为与软件需求的偏差。并不是每个软件缺陷都一定会导致软件发生故障, 缺陷只有在满足某种条件的情况下才会导致软件故障。

2. 软件测试的基本原则

1. 不完全原则 : 不完全原则表明测试是不完全的, 穷举测试是不可能的。
2. 免疫性原则 : 软件缺陷具有免疫性, 测试人员完成的测试越多, 其免疫能力就越强, 寻找更多软件缺陷也就更加困难。
3. 全程测试原则 : 全程测试原则要求软件测试不仅存在于完成程序之后, 而应该跨越整个软件开发流程。
4. 80 / 20 原则 : 80 / 20 原则是指 80% 的软件缺陷存在于软件 20% 的空间里, 软件缺陷具有空间聚集性。

3 软件测试的方法和步骤

*软件测试方法

根据测试过程是否需要运行被测试的程序,软件测试方法一般分为静态测试方法与动态测试方法。

①静态测试

静态测试是在对软件代码进行分析、检查和测试时不实际运行被测试的程序,同时它还可以用于对各种软件文档进行测试。静态测试可以采用人工检测和计算机辅助的手段进行,它适用于软件开发的全过程。静态测试方法主要有代码走通(Code Walkthrough)和 Fagan 检查两种。

②动态测试

动态测试就是通过运行软件来检验软件的动态行为和运行结果的正确性。动态测试的主要特征是计算机必须真正运行被测试的程序,通过输入测试数据,对其运行情况(即输入与输出之间的对应关系)进行分析。因此所有动态测试都必须包括两个基本要素:被测试软件和用于运行软件的数据,即测试数据。动态测试根据测试时的方法不同,分为黑盒测试与白盒测试两类。

*黑盒测试

1.黑盒测试又称为功能测试或数据驱动测试。它是在已知软件所应具有功能的前提下,通过测试来检测每个功能是否都能正常使用。

2.该方法把被测试对象看成一个黑盒子,测试人员完全不考虑程序的内部结构和处理过程,只在软件的界面上进行测试,用来证实软件功能的可操作性。

*具体检测过程:

检查程序是否满足功能要求或遗漏了功能,

程序是否能正确地接收输入数据并产生正确的输出信息,

数据结构是否错误或外部数据库访问是否错误,

界面和性能是否错误,

初始化和终止是否错误。

黑盒测试方法主要有等价类划分、边界值分析、错误推测等,它主要用于软件系统测试阶段。

*白盒测试

1.白盒测试也称结构测试或逻辑驱动测试。它是在已知程序内部结构和处理过程的前提下,通过测试来检测程序中的每条路径是否按预定要求正常运行。

2.该方法把被测试对象看成一个透明的白盒子,测试人员完全知道程序的内部结构和处理算法,并按照程序内部的逻辑测试程序,对程序中尽可能多的逻辑路径进行测试,在所有的点检验内部控制结构和数据结构是否和预期相同。

白盒测试方法主要有逻辑覆盖、基本路径测试等,它主要用于验证测试的充分性。

***软件测试过程** 通常可以将测试阶段划分成代码审查、单元测试、集成测试和系统测试 4 个阶段。

①代码审查

代码审查是一种非常有效的程序验证技术,对于典型的程序来说,可以查出 30%~70% 的逻辑设计错误和编码错误。它是由审查小组通过阅读、讨论和争议对程序进行静态测试的过程。

②单元测试

单元测试就是对软件中的基本组成单位(如一个类、类中的一个方法、一个模块等)进行测试。因为需要知道程序内部设计和编码的细节,所以单元测试一般由程序员而非测试人员来完成。通过测试可发现实现该模块的实际功能与定义该模块的功能说明不符合的情况,以及编码的错误。

③集成测试

集成测试又称组装测试或联合测试。它是指在单元测试的基础上,将模块或组件按照设计要求组装起来同时进行测试,其主要目标是发现与接口有关的问题,即模块或组件之间的协调与通信。

④系统测试

集成完模块或组件后,系统测试是确保整个测试的软件系统与系统的功能和非功能性需求保持一致。为了完成这一目的,需要开展下面几种系统测试活动:功能测试、性能测试、验收测试、安装测试。

与软件开发过程相反,测试是从模块或组件开始,自底向上逐步集成的过程。

4 软件维护的定义

*从软件生命周期角度来看

维护阶段是软件生存期中的一部分,“对软件产品进行检测,以期获得满意性能;当需要对软件产品进行修改以改正问题或对变化了的需求做出响应”。

从技术的角度,国标 GB/T 11457-89 对软件维护给出了如下定义:

软件产品交付使用后对其进行修改,以纠正故障,改进其性能和其他属性,以使产品适应改变了的环境。

5 软件维护的特点

软件维护主要分为三类

1. 改正型维护:改正软件系统中的错误,使软件能够满足预期的正常运行状态的要求而进行的维护
2. 适应型维护:使软件适应运行环境中硬软件变更的维护
3. 完善型维护:为软件增添新功能满足用户需求变更的维护

6 软件的可维护性

定义:维护人员改动和改正软件的难易程度。

提高可维护性是支配软件工程方法论所有步骤的关键目标,也是延长软件生产期的最好

的方法。

影响软件可维护的因数主要又下述三个:

1. 可理解性
2. 可测试性
3. 可修改性

提高软件可维护性的方法

1. 确定质量管理目标
2. 规范化的程序设计风格
3. 选择可维护的程序设计语言
4. 改进程序文档
5. 合理软件质量审查

7、软件再工程

再工程 (re-engineering), 也叫修理或再生, 它不仅能从已存地程序种重新获得设计信息, 而且还能使用这些信息改进或重建或重构现有地系统, 尽最大努力来提高软件整体质量。

【练习】黑盒子、白盒子测试法是常采用的测试方法, 其中白盒子测试主要用于测试 (D)。

- A、结构合理性
- B、软件功能
- C、程序正确性
- D、程序内部结构

【练习】软件测试基本方法中, 下列(D)不用测试实例。

- A). 白盒测试法
- B). 动态测试法
- C). 黑盒测试法
- D). 静态测试法

【练习】提高软件的可维护性可采取很多措施, 下列(A)不在措施之列。

- A). 提供没有错误的程序
- B). 建立质量保证制度
- C). 改进程序文档质量
- D). 明确软件质量标准

【练习】为了提高测试的效率, 应该 (D)。

- A. 随机地选取测试数据

- B. 取一切可能的输入数据作为测试数据
- C. 在完成编码以后制定软件的测试计划
- D. 选择发现错误可能性大的数据作为测试数据

■ 华图网校介绍

华图网校（V.HUATU.COM）于2007年3月由华图教育投资创立，是华图教育旗下的远程教育高端品牌。她专注于公职培训，目前拥有遍及全国各地500万注册用户，已成为公职类考生学习提高的专业门户网站。

华图网校是教育部中国远程教育理事单位。她拥有全球最尖端高清录播互动技术和国际领先的网络课程设计思想，融汇华图教育十余年公职辅导模块教学法，凭借强大师资力量与教学资源、利用教育与互联网的完美结合，真正为考生带来“乐享品质”的学习体验，通过“高效学习”成就品质人生。

华图网校课程丰富多元，涵盖公务员、事业单位、招警、法院、检察院、军转干、选调生、村官、政法干警、三支一扶、乡镇公务员、党政公选等热门考试、晋升及选拔。同时，华图网校坚持以人为本的原则，不断吸引清华、北大等高端人才加入经营管理，优化课程学习平台，提升用户体验，探索网络教育新技术和教学思想，力争为考生提供高效、个性、互动、智能的高品质课程和服务。

华图网校将秉承“以教育推动社会进步”的使命，加快网站国际化进程，打造全球一流的网络学习平台。

我们的使命：以教育推动社会进步

我们的愿景：德聚最优秀人才，仁就基业长青的教育机构

我们的价值观：诚信为根、质量为本、知难而进、开拓创新。

- 咨询电话：400-678-1009
- 听课网址：v.huatu.com（华图网校）