

# 第一章 概述

---

## 定义: 软件

必考

软件是文档、指令、数据结构的集合

- **Documents** that describe the operation and use of the programs.  
文档描述了程序的操作(operation)和使用(use)。
- **Instructions** (computer programs) that when executed provide desired function and performance;  
指令在执行时提供了所需的功能(function)和性能(performance)。
- **Data structures** that enable the programs to adequately manipulate information;  
数据结构使程序能够充分地操作信息(information)。

## 软件的3个特性

理解

1. Software is developed or engineered, it is not manufactured in the classical sense.  
软件是开发或工程的，而不是传统意义上的制造。
2. Software doesn't "wear out", but it does deteriorate.  
软件不会“磨损”，但会恶化。
3. Most software is custom-built, rather than being assembled from existing component.  
大多数软件都是定制的，而不是由现有组件组装而成。

## 定义: 软件工程

必考

The application of a **systematic , disciplined, measure-able** approach to the **development, operation, and maintenance** of software. That is, the application of engineering to software.

软件工程是将系统的(systematic)、规范的(disciplined)、可度量的(measurable)方法应用于软件的开发(development)、运行(operation)和维护(maintenance)的过程，即将工程化应用于软件中。

## 软件涉及到的人员

- consumer 需求提出者
- developer 开发人员
- user 软件的实际使用者

## 质量指标

理解

1. The Quality of the Product, **McCall**

度量软件质量的指标模型

2. The Quality of the Process, **CMM** (Capability Maturity Mode)

度量过程质量的指标模型

3. The Quality in the Context of the Business Environment, **ROI**

度量商业价值的指标模型

## 第二章 过程

---

### 定义: 过程 process

必考

process:

- A series of **ordered steps** involving activities, constraints, and resources that produce an intended output of some kind.
- 一系列涉及活动、约束和资源的步骤，这些步骤会产生某种预期的输出。

### Characteristics of process

了解

1. 过程描述了所有主要的过程活动。  
The process describes all the major process activities.
2. 这个过程使用资源，但受到一系列限制，并生产中间产品和最终产品。  
The process uses the resources, subject to a set of constraints, and produces intermediate and final products.
3. 过程可能由以某种方式链接的子过程组成。  
The process may be composed of sub-processes that that linked in some way.
4. 每个过程活动都有进入和退出标准，因此我们知道活动开始和结束的时间。  
Each process activity has entry and exit criteria, so that we know when the activity begins and ends .
5. 活动按顺序组织，因此一个活动相对于另一个活动的执行时间很清楚。  
The activities are organized in a sequence, so that it is clear when one activity is performed relative to the other activity .
6. 每个过程都有一组指导原则来解释每个活动的目标。  
Every process has a set of guiding principles that explain the goals of each activity.
7. 约束或控制可以应用于活动、资源或产品  
Constraints or controls may apply to an activity, resource, or product.

### 定义: 生命周期 Life cycle

必考

- 当过程涉及到建设某些产品，我们有时把过程称为生命周期。
- when the process involves the building of some product, we sometimes refer to the process as a life cycle.

- The life cycle of a software product include **conception**(概念化), **implementation**(实现), **delivery**(交付), **use**, **maintenance**.(维护)

## 定义: 模型

了解

- 一个过程可以用多种方法来描述, 使用文字、图片或组合

A process can be described in a variety of ways, using text, pictures or a combination.

## 瀑布模型

必考(步骤与优缺点)

### Step (8)

1. Requirements Analysis 需求分析
2. System Design 系统设计
3. Program Design 程序设计
4. Coding 编程
5. Unit & Integration Test 单元&集成测试
6. System Test 系统测试
7. Acceptance Test 验收测试
8. Operation & Maintenance 使用与维护

特点

1. One of the first process development models  
是第一个被提出的开发模型之一
2. Works for well understood problems with minimal or no changes in the requirements  
仅需极少的需求或没有任何变化即可解决众所周知的问题
3. It presents a very high-level view of the development process开发过程的顶层理解
4. and sequence of process activities 过程活动的顺序
5. Each major phase is marked by milestones and deliverables 每个主要阶段都具有里程碑和可交付成果

## 优点

- The waterfall model can be very useful in helping developers lay out what they need to do.  
在帮助开发人员规划他们需要做的事情时非常有用
- Its simplicity makes it easy to explain to customers who are not familiar with software development.  
它的简单性使得向不熟悉软件开发的客户解释它很容易
- It makes explicit which intermediate products are necessary in order to begin the next stage of development.  
它明确了哪些中间产品是必要的, 以便开始下一阶段的开发。
- Many other, more complex models are really just embellishments of waterfall, incorporating feedback loops and extra activities.

许多其他更复杂的模型实际上只是瀑布的点缀

## 缺点

- Provides no guidance how to handle changes to products and activities during development (assumes requirements can be frozen)

没有提供有关如何在开发过程中处理产品和活动更改的指导（假设要求可以冻结）

- Views software development as manufacturing process rather than as creative process

将软件开发视为制造过程而非创意过程

- There is no iterative activities that lead to creating a final Product

没有迭代活动可以创建最终产品

- Long wait before a final product

长时间等待最终产品

## 第三章 项目管理

---

### 定义: 进度 Schedule

- A project schedule describes the software development cycle for a particular project by enumerating the phases or stages of the project and breaking each into discrete tasks or activities to be done.
- 项目进度通过列举项目的时期或阶段，并将每个阶段分解为待完成的离散任务或活动来描述特定项目的软件开发周期。
- The schedule is a **timeline** that shows when activities will begin and end, and when the related development products will be ready.
- 进度是显示活动开始和结束的时间线，以及相关开发产品准备就绪的时间线

### 定义: 活动

- An activity is a part of the project that takes place over a period of time.
- 活动是在一段时间内发生的项目的一部分

### 定义: 里程碑 Milestone

- A milestone is the completion of an activity, a particular point in time.
- 里程碑是一个活动的完成，一个特定的时间点。

## WBS

- 工作分解结构(WBS)将项目描述为一组离散的工作片段。
- work breakdown structure depicts the project as a set of discrete pieces of work.
  - Precursor is an event or set of events that must occur before the activity can begin.  
前驱是在活动开始之前必须发生的一个或一组事件。
  - Due date is the date by which the activity must be completed, frequently determined by contractual deadlines.  
截止时间是活动必须完成的日期，通常由合同期限决定。
  - Duration is length of time needed to complete an activity.  
工期是完成一项活动所需的时间长度。
  - Endpoint is usually a milestone or deliverable  
终点是里程碑或可交付件。

## 关键路径

- The critical path is the one for which the slack at every node is zero.
- 关键路径是每个节点的松弛度为零的路径。

从活动图找出关键路径!!! 书后两条练习题 ☆

```
1 //对于活动A_B, P-A-B-S 表示前驱活动P_A, 后继活动B_S
2 //活动A_B可能有不止一个前驱/后继, 此时用Pi_A, B_Si表示
3 #define time_point int //时刻
4 #define time_period int //时间段
5 #define START 1 //开始时间=1
6 #define END END //结束时间END
7 int pre(A_B){
8     //返回活动A_B的前驱活动数量
9 }
10
11 int sub(A_B){
12     //返回活动A_B的后继活动数量
13 }
14
15 time_period T(A_B){
16     //返回活动A_B需要的时间
17 }
18
19 time_point EST(A_B){
20     time_point re;
21     int preNum = pre(A_B);
22     if(preNum == 0)
23         re = START;
24     if(preNum == 1)
25         re = EST(P_A) + T(P_A);
26     if(preNum > 1)
27         re = max(EST(Pi_A) + T(Pi_A));
28     return re;
29 }
30
31 time_point LST(A_B){
32     time_point re;
33     int subNum = sub(A_B);
34     if(subNum == 0)
35         re = END - T(A_B);
36     if(subNum == 1)
37         re = LST(B_S) - T(A_B);
38     if(subNum > 1)
39         re = min(LST(B_Si) - T(A_B));
40     return re;
41 }
```

## 第四章 需求分析

### 定义: 需求

- An **Expression** of desired **Behavior**

期望行为的一种表达。

#### 四种需求

- functional requirement : 功能性需求  
describes required behavior in terms of required activities
- nonfunctional requirement (quality function) 非功能性需求(质量需求)  
describes some quality characteristic that the software must possess.
- design constraint 设计约束  
a design decision,such as choice of platform.
- process constraint 过程约束  
a restriction on the techniques or resources that can be used to build the system.

## ERD (实体关系图)

- entity-relationship diagram (ERD):  
is a popular graphical notational paradigm.
- 三要素
  - An entity, epicted as a rectangle, represents a collection of real-world objects that have common properties and behaviors.
  - A relationship, depicted as an edge between two entities, with diamond in the middle of the edge specifying the type of relationship.
  - An attribute, an annotation(注释) on an entity that describes data or properties associated with the entity.

## DFD Data Flow Diagram数据流图

Process, data flow, data store, actors

## UCD (用例图)

- 概念: use-case diagram(UCD) is similar to a top-level data-flow diagram that depicts observable,user-initiated functionality in terms of interactions between the system and its environment.用例图(UCD)类似于顶级数据流图，根据系统和环境之间的交互描述可观察的、用户发起的功能。
- 用处: 用于表示需求,不是设计
- 怎么画?
  - 边界
  - case

## 两种文档 (区别作用)

- require **definition** 需求分析说明书 For **Customer**  
一般是对某个市场或者是客户群来讲的，类似于调研报告，重点是体现出产品要满足哪些功能，哪些是重点、热点。
- require **specificatiuon** 需求规格说明书 For **Developer**

是从业务规则讲起的，细一点偏向于软件的概要设计。是从开发、测试的角度去讲产品功能，里面要包含原型界面、业务接口、活动图等

## 第五章 体系结构

---

### 定义: 设计

- Design is the creative process of figuring out how to implement all of the customer's requirements;
- the resulting plan is also called the design.
- 设计是找出如何实现客户所有要求的创造性过程;
- 最终的方案也称为设计。

### 体系结构三要素(3C) ☆

软件体系结构={构件, 连接件, 约束}

- component(构件): 可以是一组代码, 如程序模块, 也可以是一个独立运行的程序, 如数据库服务器。
- connector(连接件): 可以是过程调用, 管道, 远程调用等, 用于表示构建之间的相互作用。
- constraint(约束): 对象的连接规则。

分解

自顶向下

合成

### Style

概念!

- 管道过滤器 **Piper-and-Filter**
  - 管道是干什么的? 过滤器是什么?
  - Transmit data form one filter to then next
  - MIS system of XIDIAN University 西电迎新系统
- 客户服务器 **Client-Server**
  - Most transaction processing system(事务处理系统) Eg. Bank System
- ...

## 第六章 模块设计

---

什么是模块化?

- Modularity, also called separation of concerns, is the principle of keeping separate the various unrelated aspects of a system, so that each aspect can be studied in isolation.
- 不相关的方面的相互独立

什么是独立性? Independence? Separation of Concern

### Coupling 六耦合

- 非直接耦合 Uncoupled
- 数据耦合 Data
- 标记耦合 Stamp
- 控制耦合 Control
- 公共耦合 Common
- 内容耦合 Content

## Concerns 七内聚

- 偶然内聚 Coincidence
- 逻辑内聚 Logic
- 时间内聚 Temporal
- 过程内聚 Procedural
- 通信内聚 Communicational
- 功能内聚 Functional
- (信息内聚) Information

耦合与内聚 各考一个(判断是什么类型的耦合\内聚)

## 第七章 编程

---

### 程序三要素

- 控制结构 Control Structural
- 算法 Algorithm
- 数据结构 Data Structure

### 注释: Head\_comment\_Block

里面有什么东西(7个)

1. Program SCAN
2. Programmer
3. Calling sequence
4. Version
5. Revision
6. Purpose
7. Data structure
8. Algorithm

## 文档

内部 Internal documentation

- Head Comment Block
- Other Program Comment
- Meaningful Variable Names and Statement Labels
- Formatting to Enhance Understanding

外部 External documentation

- describing the problem



- describing the algorithm
- describing the data

各包括什么content

## 第八章 单元测试 ☆

---

测试方法

测试用例:

- A test point or test case is a particular choice of input data to be used in test a program.
- 测试点或测试用例是用于测试程序的输入数据的特定选择

覆盖 由方法选数据

黑盒白盒: 各测试什么(功能/结构)

- We view the test object form the outside as a closed or black box whose contents are unknown, our testing **feeds inputs to the closed box and notes what output is produced.**  
我们从外部将测试对象视为一个关闭的或内容未知的黑盒，我们的测试将输入输入到关闭的盒子中，并记录产生了什么输出。
- We view the test object as an open box and or white box, we can use the **structure** of the test object to test in different way.  
我们将测试对象看作是一个开放的盒子或者白盒子，我们可以使用测试对象的结构以不同的方式进行测试。

路径测试-----给程序, 画流程图,画控制图; 给图, 找测试路径

## 第九章 系统测试

---

Rush Time 高峰时间

概念: 前五个

- 压力 Stress Test
- 容量 Volume Test
- 配置 Configuration Test
- Compatibility Test
- Regression Test
- ...

## 第十章 系统交付

---

概念

- R 可靠性  
The software reliability is a **possibility** of the system will **operate without failure** under a given conditions and a given time interval.  
软件可靠性是指系统在给定条件和给定时间间隔下无故障运行的可能性。
- A 可用性  
The software availability is the **probability** that a system **is operating successfully** according to specification at a given point of time.

- M 可维护性

The software maintainability is the probability that , for a given condition of use, a maintenance activity **can be carried out** within stated time interval and using stated procedures and resources.

给定的使用条件下，可以在规定的时间间隔内并使用规定的过程和资源进行维护活动的可能性。

- PPT原话!!!!

## 第十一章 系统维护

---

### 维护的定义

Any work done to change the system after it is in operation is considered to be maintenance.  
在系统运行后为改变系统所做的任何工作都被认为是维护。

四类维护: 判断是什么样的维护...

1. Corrective Maintenance : Bug 改正性维护
2. Adaptive Maintenance : Table-Driven Design 适应性维护
3. Perfective Maintenance : DB Upgrade to New Version 完善性维护
4. Preventive Maintenance : Printer Driver: Print-line 预防性维护

笔记制作: 18级胡博

欢迎来我的[个人博客](#)玩

希望你看到的时候这个博客还在

西电无纸化之软件工程分群:929629717