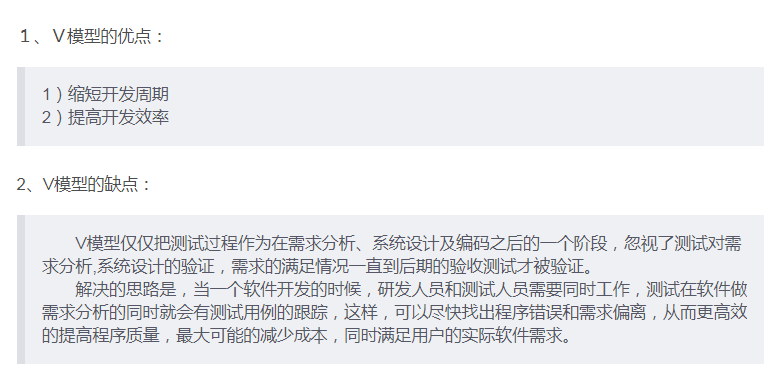
本文件整理的考点绝对绝对绝对不全，这里只整理了往年简答题中的知识点，**欢迎补充**

1、易考的Process Model（要学会文字描述每一种模式）

<https://blog.csdn.net/sandalphon4869/article/details/92843264>

补充：V model



2、几种时间/路径的求法

Critical path关键路线（考虑：万一活动图有虚线怎么做，见课本例题）：

从开始到结束得所有路径中，所用时间最长的为关键路径（有可能有多条相同的）

最早开始时间（以前者为基准）：

在关键路径上，从开始到该任务的最早执行的时间

**第一个结点（开始结点）的最早执行的时间是1，不是0**

其他结点的最早开始时间为：[从开始结点到Activity前者结点经过的最长的时间总长]+1

最晚开始时间（以前者为基准）：

关键路径的总时间-反向得出该任务的时间

关键路径上的Activity（前者、后者都在）的最晚开始时间是同最早开始时间。

其他结点的最晚开始时间为：关键路线的总时长 - [从结束结点到Activity前者结点经过的最长的时间总长]+1

Slack time松弛时间：

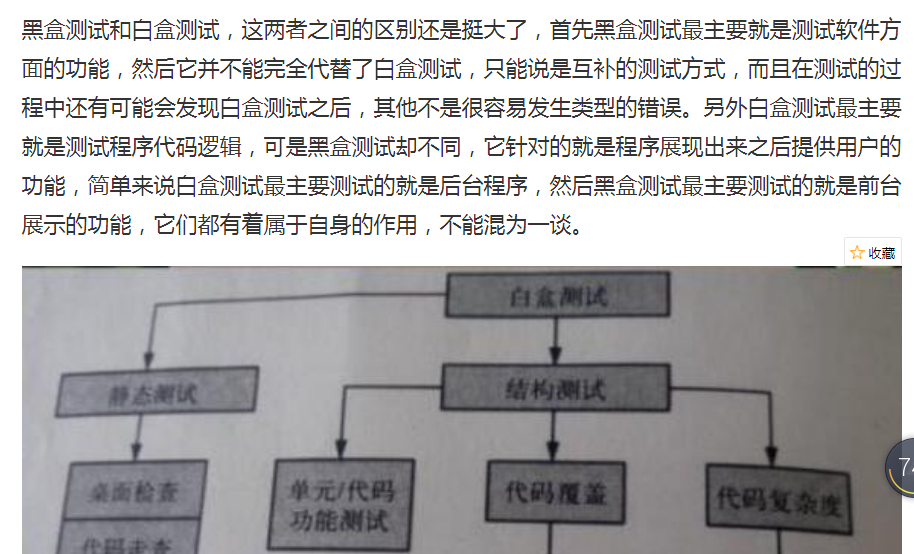
第一种求法：最晚开始时间-最早开始时间

第二种求法：关键路径的总时间-包含该任务的关键路径花的时间

**关键路径上的Activity（前者、后者都在）的Slack time为0**

Precursors前驱结点：  
是Activity（前者-后者）中前者的前驱结点

1. 黑盒测试与白盒测试的内容和特点



（1）黑盒测试是最重要的一类软件测试方法。黑盒测试仅需知道被测对象的输入和预期输出，不需要了解其实现的细节。典型的黑盒测试方法包括**等价类测试（Equivalence Class Partitioning）、边界值测试、基于决策表**的测试方法等，可从如下方面来评价某种测试方法的质量：

<https://blog.csdn.net/tlonline/article/details/46470863>

1. 白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试的测试方法有代码检查法、静态结构分析法、静态质量度量法、逻辑覆盖法、基本路径测试法、域测试、符号测试、路径覆盖和程序变异。

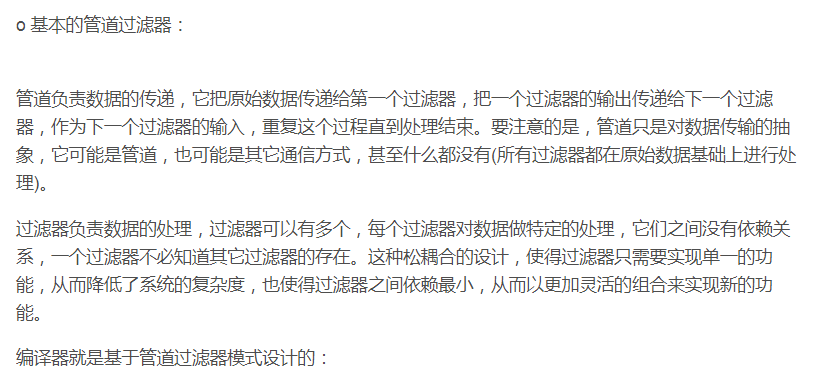
<https://blog.csdn.net/tlonline/article/details/46470449>

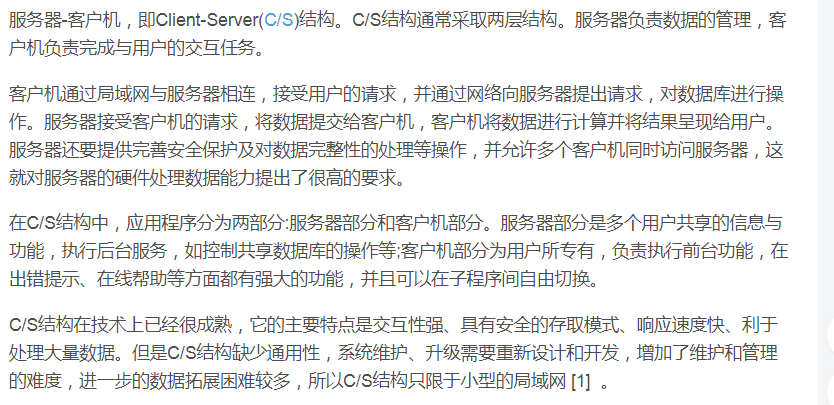
1. Alpha与Beta测试

它们都是验收测试！前者为内测，后者为公测。

https://blog.csdn.net/weixin\_43214609/article/details/94735436?utm\_medium=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control&depth\_1-utm\_source=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control

1. 五种设计模式





Layer模式：<https://blog.csdn.net/trecn001/article/details/93858852>（详见课本）

发布/订阅模式：<https://www.jianshu.com/p/c391c77a8771>

Repositories模式：<https://blog.csdn.net/cangqiong_xiamen/article/details/106034387>

6、耦合与内聚类型的区分

<https://blog.csdn.net/guangod/article/details/102456301>

1. 软件的可靠性、可用性、可维护性

<https://blog.csdn.net/sandalphon4869/article/details/92844182>

补充：Availability（可用性）= MTBF/(MTBF+MTTR)，要**理解这几个量的含义**

8、McCabe度量法

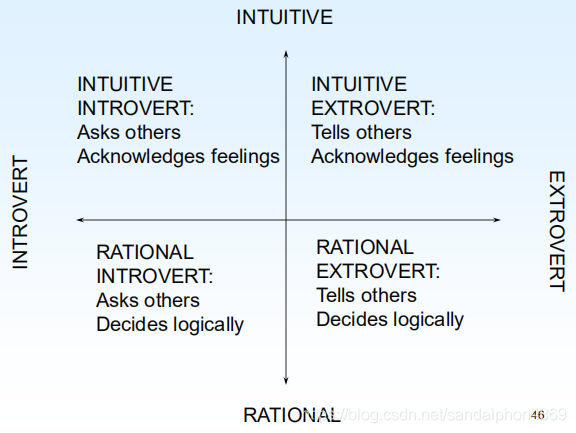
McCabe度量法是一种基于程序控制流的复杂性度量方法。McCabe复杂性度量又称环路度量（Cyclomatic Number）,其计算公式为: V(g)=m-n+2,其中m和n分别代表图中的边数和顶点数。

9、内向者（Extrovert）与外向者（Introvert）

前者乐于提供建议；后者寻求建议

1. 理性者（Rational）与直觉者（Intuitive）

前者检验事实、分析所有可能性；后者基于感觉和情感



1. UML图的种类、区分（要了解几个常见UML图的定义和例子）

<https://blog.csdn.net/itfly8/article/details/100890362>

1. 软件测试的步骤及内容

<https://blog.csdn.net/sandalphon4869/article/details/92844078>

单元测试：测试单个模块；  
集成测试：测试集成的模块；  
系统测试：测试整个软件系统；

其中系统测试主要的四个步骤：

1.功能测试：检查集成的系统是否按照需求中指定的那样执行它的功能

2.性能测试：将集成的构件与非功能需求进行比较

3.验收测试：客户参与的测试，目标是确保系统符合他们对需求的理解

4.安装测试：在实际运行环境中进行的测试

1. 回归测试

**定义：**回归测试是用于新的版本或发布的一种测试，以验证与旧版本或发布相比，它是否仍然以同样的方式执行相同的功能。  
**原因：**由于在测试过程中可能在修复已有故障的同时引入新的故障，故需要进行回归测试。

1. 集成测试的四种模式

见12中的链接。

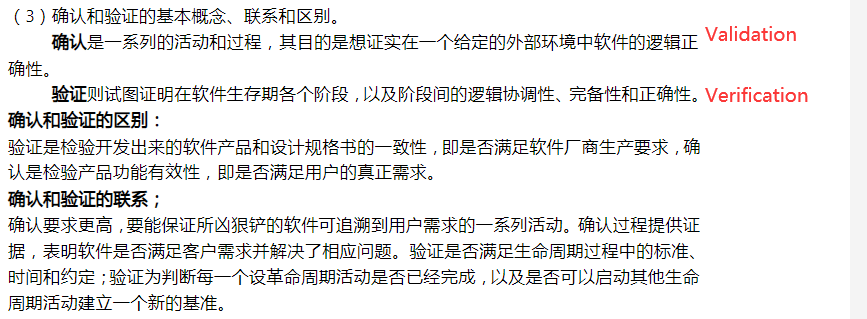
1. Quality

1.The quality of the product：McCall’s quality model

2.The quality of the process：CMM、ISO 9000、SPICE

3.Quality in the context of the business environment：ROI (return on investment)

1. 确认与验证



1. 两种项目管理结构

Chief programmer team 和 egoless approach，分别是集权和民主的管理模式

18、不同角色的作用

**Briefly describe the roles of analyst, designer, programmer, tester, and trainer.**

1.The requirement analyst work with customer, break down what the customer wants into discrete requirement.

2.The designer generate the system level description of what the system is to do.

3.The programmer write line of code that implement what requirement specify.

4.The tester catches faults that programmer overlook.

5.The trainer shows users how to use the system.

19、好的设计的特点

1. 好的软件的特点
2. 好的系统的特点
3. DDF/ER/Event Trace图/State Machine

**DDF图：数据流图，对功能以及从一个功能到另一个功能的数据流建模**

1. **R图：表示概念模型的一种方法，包含实体、联系和属性三个要素**

**Event Trace：事件踪迹，关于现实世界实体之间交换的事件序列的图形描述**

**State Machine：状态机，描述了系统与环境之间的所有对话**

23、Pairs

If there is n worker in project, then there are**n(n-1)/2** pairs of communication.

24、维护的四种类型



【重点的定义】

**Software：**Software is a set of programs (Instruction), data (structures) and documents

**Software Engineering：**Software engineering means the application of a systematic, disciplined and measurable approach to the development, operation, and maintenance of software. That is, the application of engineering to software.

**Process:** A set of ordered tasks (a series of steps, including activities, constraints and resources)

**Program:** A defined set of algorithms or processes that are completed in order to achieve a task

**System:** A collection of a set of entities, a set of activities, a description of the relationships among entities and activities, and a definition of the boundary of the system

**System的要素：**Activities and Objects, Relationships and the System Boundary

**Life Cycle：**When the process involves building of some products , we sometime refer to the Process as a life cycle.

**Software Life Cycle：**软件从产生（提出）直到报废或停止使用的生命周期

The software development usually involves **requirement analysis and definition, system design, program design, program implementation, unit testing, integration testing, system testing, system delivery and maintenance.**

**【中文翻译】**

软件生命周期的六个基本步骤：

* 制定计划
* 需求分析
* 设计
* 程序编码
* 测试
* 运行维护

**软件工程三要素：工具、方法、步骤（四要素则加上“范式”）**

**Project Schedule：**A time timeline that shows when activities will begin and end, and when the related development products will be ready

**Activity：**A part of the project that takes place over a period of time

**Milestone：**The completion of an activity

**活动四要素：**precursor, duration, due date, and endpoint

**Risk：**An unwanted event that has negative consequences

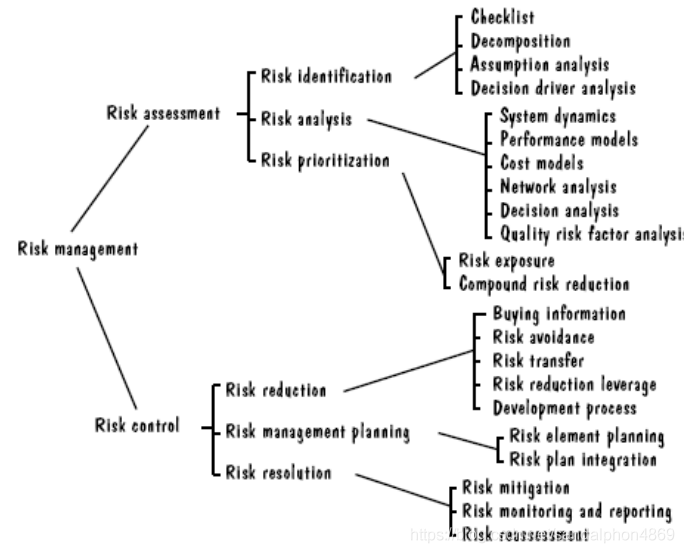
**分为Risk assessment风险评估和Risk control风险控制**

**Risk assessment风险评估：**

* **Risk identification风险定义**
* **Risk analysis风险分析**
* **Risk prioritization风险优先次序**

**Risk control风险控制：**

* **Risk reduction风险减少**
* **Risk management planning风险管理计划**
* **Risk resolution风险解决**



**Project Plan**

四种Test Documents

**Test Plan**

**Test Specification and Evaluation**

**Test Description**

**Test Analysis Report**

**19级学长有话说：这里是后话，考完以后发现整理的重点大部分都没考，考的反而是一些看起来细枝末节的知识点，┭┮﹏┭┮**