

软件工程重点总结 2018.01

本文档完全根据老师所画内容进行整理，其中有部分作者自己的理解，如有错误，欢迎指出，谢谢。

1. 什么是软件工程：

软件工程是将系统化、规范化、可度量的方法应用于软件开发、运行和维护过程，即将工程化应用于软件中的方法的研究。从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

2. 什么是软件危机：

落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题的现象。如：① 开发成本和进度的估计常常不准确；② 用户对交付的软件系统不满意的现象经常发生；③ 软件质量无保证、可靠性差；④ 软件可维护性非常低；⑤ 软件通常没有适当的文档资料；⑥ 软件成本在计算机系统总成本中所占比例逐年上升；

3. 什么是软件：

软件是计算机程序，规程以及运行计算机系统过程需要的相关文档和数据。在软件工程中，软件=程序+文档+数据。

4. 软件的生命周期（即开发所经历的阶段）：

软件生命周期是软件产品从考虑其概念开始到交付使用，直至最终退役为止的整个过程。分为如下三个阶段：

① 软件定义：软件定义的任务是确定待开发的软件系统要做什么，又称为系统分析，可进一步划分为问题定义，可行性研究，需求分析三个阶段。

② 软件开发：软件开发的任务是具体设计和实现软件。通常由下面四个阶段组成：概要设计，详细设计，编码和单元测试、综合测试，前两个统称为系统设计，后两个统称为系统实现。

③ 软件运行和维护：运行维护的任务是根据软件运行中的问题，对其进行修改，是系统能持久的满足用户需求。

5. 软件过程模型：

模型就是对实际事物、实际系统的一种抽象。所以软件过程模型是软件开发的全部过程、活动和任务的结构框架，直观的表达了软件开发的全过程，明确规定了要完成的主要活动、任务和开发策略。

作用：告诉人们应该去遵循一个什么样的过程和策略去开发软件系统。

关于以下原型的优缺点简单记住几条，考试扯扯就行了，主要理解各模型的过程。

1) 瀑布模型：这是个重点，老师画重点一直在说这个

瀑布模型包含了各项软件工程活动，即制定开发计划、进行需求分析、软件设计、程序编码、测试及运行维护。它规定了各项软件工程活动自上而下、相互衔接的固定次序。瀑布模型是经典，是软件过程模型的里程碑。

特点：

- ① 上一项活动接受该项活动的工作对象作为输入；
- ② 利用这一输入实施该项应完成的内容；
- ③ 产生本项活动的相关产出，作为输出传给下一项活动；
- ④ 对本项活动执行情况进行平生。如果活动执行得到确认，则继续进行下一项活动；

否则返回前一项，甚至对前一项的活动进行返工。

缺点：

- ① 在软件开发的初始阶段指明全部需求是非常困难的，有时甚至是不现实的。
- ② 需求一旦确定，用户和软件项目负责人需要等一段时间（经过设计、实现、测试、运行等阶段）才能得到一份软件的最初版本。
- ③ 瀑布模型的软件活动是文档驱动的，当阶段之间规定过多的文档时，会大大增加软件开发的工作量，而且以文档的完成情况来评估项目完成进度时，往往会产生错误的结论。

2) 原型模型：这也是个重点，老师画重点一直在说这个

原型表示软件的一个早期可运行的版本，反映最终系统的部分重要特征。通俗的讲，原型模型就是先获取用户的基本需求，然后开发一个小型软件系统原型，然后根据用户的反馈，对原型进行不断的修改。

作用：准确地确定用户的需求。

使用原型的策略：

- ① 废弃策略：先构造一个功能简单且性能要求不高的原型系统，根据用户的反馈，反复分析改进，形成好的设计思想，据此设计出完整、准确、一致、可靠的最终系统。原系统废弃不用。
- ② 追加策略：先构造一个功能简单且性能要求不高的原型系统，作为最终系统的核心，然后通过不断地进行扩充修改，逐步追加新要求，最后发展成为最终系统。

优点：

- ① 用户参与，尽早揭示软件中可能存在的风险及不确定因素，尤其是关于用户需求一致性方面的风险。
- ② 开发过程与用户培训过程同步，系统易维护，对用户更友好，产品柔性好。

缺点：

- ① 不经过系统分析并对系统进行整体划分，直接用原型模拟系统功能比较困难。
- ② 对于计算量大、逻辑性较强的程序模块，原型法很难真正构造出来供用户评价。
- ③ 对于批处理系统，其大部分处理是内部进行的，应用原型法有一定困难。
- ④ 项目文档容易被忽略，给后期原型的改进和维护造成困难。
- ⑤ 对丢弃策略，很多工作被浪费掉，增加开发成本，降低开发效率。

以下模型划重点时一语带过，非重点。

3) 螺旋模型

思想：使用原型及其他方法来降低风险。

4) 喷泉模型

比较灵活，具有迭代性和无间隙性。可以随时回到上一个阶段。

5) 增量模型

可以让客户把需求逐步提出来。

6. 软件过程和面向对象，面向过程的区别：画重点的时候提出来的无厘头的问题

软件过程是指软件开发所经历的一系列阶段，而面向对象和面向过程只是这些阶段中某个阶段所采用的一些方法而已。（随便扯扯，有更好的回答欢迎反馈）

7. 可行性研究是干什么的？

可行性研究是研究这个软件项目是否值得去开发，需要从项目的技术可行性、经济可行性、操作可行性、社会可行性等方面进行研究和分析，并最终做出该项目是否具有可行性的结论。

8. 需求工程：

需求工程是指应用已证实有效的技术、方法进行需求分析，确定客户需求，帮助分

析人员理解问题并定义目标系统的所有外部特征的一门学科。

如果您认为本文档对您的复习有些许的帮助，整理成本费 1-2 元，请打赏，谢谢。



9. 需求开发过程：

- ① 需求获取：通过访谈，问卷调查，专题讨论会等手段获取用户需求。
- ② 需求建模：用清晰、简明的方式将需求分析获得的信息记录下来，得到一个逻辑模型。
- ③ 需求规格说明：用户和开发人员都充分了解用户需求后，将共同的理解以规范化的形式准确的表达出来，形成需求规格说明书。
- ④ 需求评审：在将需求交付设计之前，对需求规格说明书进行彻底的检查和修改。

10. 需求管理过程：

软件开发过程中，可能会有需求的提出和更改，所以要进行需求管理。

- ① 需求变更控制：当出现需求变更时，就要对变更影响和成本进行分析，通过用户方和开发方组成的变更控制委员会来决策以规范和控制需求变更（肯定不能一方说改就改）
- ② 需求版本控制：它保证在需求文档中记录和反映所有的需求变化。
- ③ 需求跟踪：维护需求的可跟踪性信息（哪个需求谁做了等等）。
- ④ 需求状态跟踪：跟踪需求的状态（哪个需求进行到什么程度了，已完成还是已验证等等）

11. 模块化和模块独立性：

模块化是指解决一个复杂问题时自顶向下逐层把软件划分成若干模块的过程，每个模块完成一个子功能。

模块独立性是指软件系统中每个模块只涉及软件要求的具体的子功能，而和软件系统中其他的模块的接口是简单的。

12. 面向对象的基本概念：

- ① 对象：将客观世界的实体抽象为问题空间中的对象。
- ② 类：把具有相同特征和行为的属性归在一起就形成了类。
- ③ 消息：实现对象类之间的通信和任务传递。
- ④ 封装：把对象属性和操作结合在一起，构成独立的单元，只能通过有限的接口于对象发生联系，不能直接存取对象的属性。

- ⑤ 继承：子类可以自动拥有父类的全部属性和操作。
- ⑥ 多态性：子类对象可以像父类对象一样使用，同样的消息可以发给父类也可以发给子类。

13. UML：

UML，统一建模语言，以面向对象图的方式来描述任何类型的系统模型。

14. 识别分析类：

- ① 识别实体类：实体类用于描述必须存储的信息，同时描述相关的行为。
- ② 识别边界类：边界类在系统与外界之间，为他们交换各种信息与事件。功能：输入、输出和过滤。
- ③ 识别控制类：控制类与业务过程相关，它们控制整个业务的流程和执行次序。是一组操作的集合，用来协调各个边界类对象、实体类对象等。

15. 用户界面设计原则：

- ① 置用户于控制之下；② 减轻用户的记忆负担；③ 保持界面一致；

16. 软件测试：

所有的软件问题都称为软件缺陷，不仅仅指 BUG。软件测试是为了发现软件缺陷，而无法证明软件的正确性。

- ① 静态黑盒测试：通过看产品说明书来查找缺陷。（好奇葩的方式）
- ② 动态黑盒测试：不追究代码细节，通过使用软件来进行测试。
- ③ 静态白盒测试：通过看说明，结构，代码来找缺陷。
- ④ 动态白盒测试：利用查看代码功能和实现方式得到信息，来确定如何展开测试。

17. 软件测试策略：

- ① 单元测试：针对软件设计的最小单元程序模块进行测试的工作。
- ② 继承测试：按照一定的策略对单元测试的模块进行组装，并在组装过程中进行模块接口与系统功能测试；
- ③ 确认测试：目的是验证软件的有效性。按照软件需求说明书对软件的功能和性能要求进行测试。
- ④ 系统测试：为了测试软件安装到实际应用的系统中后，能否正常工作，以及对系统运行出现的各种情况的处理能力。

18. 软件维护：

软件维护是为了纠正错误或满足新的需求而修改软件的过程。大致分为：

- ① 纠错性维护；② 适应性维护；③ 完善性维护；

19. 逆向工程和再工程：

- ① 逆向工程：分析已有的程序，寻求比源代码更高级的抽象表示形式，比如通过反汇编、反编译和动态跟踪等方法，分析一个软件的实现过程，这种行为就是逆向工程。
- ② 再工程：是指在逆向工程所获得的信息的基础上修改重构已有的系统，产生一个新的版本。

以上内容为老师所画重点的全部内容，但是第十章未进行整理，老师提到的是 10.2、10.5 和 10.8 节，追求高分的同学请自行理解。

至于各种画图，这里不再总结，可以去找往年的假题，练习一下画图等等。