

学习体会：高教自考软件工程课程概说总结 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/294/2021_2022__E5_AD_A6_E4_B9_A0_E4_BD_93_E4_c67_294867.htm 未接触软件工程之前一直都很想学这门课程，因为觉得这门课很NB，是那些有工程师称号的高手才摆弄的东西。但学过之后，最大的感触却是：软件工程方法一定要从娃娃抓起，否则到了后面坏习惯已经养成后再回过头来修正，那绝对是地狱般的磨难。下面就是我在近两个月的学习中一些总结和体会，希望对后来者有所补益。由于是初学这门课程，难免浅薄和有所错漏，还望大家多多指教。软件工程的由来 据说上个世纪60年代的程序员都是天才，写程式就像写日记一样，吃过晚饭没事干随手就可以写几个出来玩，第二天还可以拿去卖钱。所以那时候程序员在大家眼中，跟那些搞美术，音乐的是一类的，被称为“艺术家”。但事过境迁，就像任何人都不会嫌钱多一样，永远都不会有人嫌CPU快的。于是，随之而来的就是硬件的迅猛发展和越来越变态的软件。记得以前常去同学家拷游戏，通常几张软盘就可以搞定，而现在的游戏，两三张CD-ROM都算少的了。像如此庞大复杂的怪物，就算你是如何的天才，一个人肯定是搞不定的，否则，等你把程式写出来，人家Intel连奔腾N都开发出来了。既要开发大型的软件还要追求速度(这样才能赚钱)，于是很自然地，合作的概念被提了出来。在开始合作的初期，由于大家都习惯了当很有个性的“艺术家”，结果可想而知，一个是毕加索派的，而另一个是意大利印象派的，再加上一个画泼墨山水画的，要是像这样凑出来的东西都能不出问题的话，那么Bill早就转行

了。所以，那时候的大型软件，据说“蓝屏”比WINDOWS 98还多。马克思告诉我们，万物都是从量变到质变的。随着问题的不断涌现，一些master们开始尝试去总结经验，并归纳了一些规范去指导软件的分析，设计，实现，测试，维护，人员交流协作，项目预算及时限控制等方方面面，这就是软件工程的前身。软件工程到现在已发展了30多年，可以说是相当成熟的了。现在开发软件，据说都是一大帮人排排坐，按着一整套的规章制度来干活。于是，软件开发成了“工程”，程序员也就沦为“工人”了。最初提出问题的是Dijkstra. 他于1968年写给ACM的一封信题为Goto Statement Considered Harmful的信中，指出了GOTO语句的负面作用，并提出了解决之道，其引发的一系列效应最终带来了软件工程的诞生。软件的核心 无论是在上个世纪还是在现在，软件开发所涉及的工作基本上都没有变化，它们都起始于一个实际需要或某个灵感，然后就是分析，设计，编码，调试，维护。这些任务以某种方式动态地结合起来就构成了软件开发的整个过程，这就是所谓的“软件开发周期”。但对于这些工作，具体怎样做，什么时候做，每个人都有自己的一套方式，甚至有的人有几套方式。这样，当几个人合在一起干活的时候，最终的结果就只能是一片混乱。所以需要一套规则，大家都按规则来办事，问题就会少得多。好的规则就叫做规范，规范都是由一些master们根据经验总结的，又经过长时间的历练，不断地被补充修正，可以说都是精华，按照规范来干活，对于提高软件质量和工作效率自然大有帮助。而软件工程，说白了，就是这样一套用于软件的开发，以提高软件质量和程序员工作效率为目的的规范。其核心就是，对于

软件开发的5个重要组成部分：需求分析，设计，编码，调试，维护，如何组织这5个部分的工作，以及如何完成每一个工作。简单来说，就是对于总体的组织和对于局部的实现。规范只是提供一个好的例子，以描述一种思想，具体到每一个环节怎样实现，对于不同的公司或团体则是各有千秋，因为根本就不可能存在一套放之天下皆可行的标准。就像C，也只是提供了一套标准，不同的编译器都有各自的实现，对标准的支持程度也互不相同。所以，在不同的公司或团体中，尽管核心思想都是大同小异，但具体到每一个步骤，往往都是不相同的。我手上就有一份GB8567-88的文档模板，对于那些顶多只有几千行的小程序来说，假如真按上面的要求全写上了，简直就是一种折磨！据说，当前业界最权威的标准是CMM. 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com