1. **什么是软件危机？它有哪些表现？为什么会出现软件危机?**

软件危机是指在计算机软件开发、使用与维护过程中遇到的一系列严重问题和难题。它包括两方面:如何开发软件,已满足对软件日益增长的需求;如何维护数量不断增长的已有软件。

软件危机的典型表现:(1)对软件开发成本和进度的估计常常很不准确。常常出现实际成本比估算成本高出一个数量级、实际进度比计划进度拖延几个月甚至几年的现象。而为了赶进度和节约成本所采取的一些权宜之计又往往损害了软件产品的质量。这些都降低了开发商的信誉,引起用户不满。(2)用户对已完成的软件不满意的现象时有发生。(3)软件产品的质量往往是靠不住的。(4)软件常常是不可维护的。(5)软件通常没有适当的文档资料。文档资料不全或不合格,必将给软件开发和维护工作带来许多难以想象的困难和难以解决的问题。(6)软件成本、软件维护费在计算机系统总成本中所占比例逐年上升。(7)开发生产率提高的速度远跟不上计算机应用普及的需求。

软件危机出现的原因:(1)来自软件自身的特点:是逻辑部件,缺乏可见性;规模庞大、复杂,修改、维护困难。(2)软件开发与维护的方法不当:忽视需求分析;认为软件开发等于程序编写;轻视软件维护。(3)供求矛盾将是一个永恒的主题:面对日益增长的软件需求,人们显得力不从心。

1. **假设自己是一家软件公司的,总工程师，当把图 1.1给手下的软件工程师们观看，告诉他们及早发现并改正错误的重要性时，有人不同意这个观点,认为要求在错误进人软件之前就清除它们是不现实的,并举例说：“如果一个故障是编码错误造成的,那么，一个人怎么能在设计阶段清除它呢？”应该怎么反驳他？**

答:在软件开发的不同阶段进行修改付出的代价是很不相同的,在早期引入变动,涉及的面较少,因而代价也比较低;在开发的中期,软件配置的许多成分已经完成,引入一个变动要对所有已完成的配置成分都做相应的修改,不仅工作量大,而且逻辑上也更复杂,因此付出的代价剧增;在软件“已经完成"是在引入变动,当然付出的代价更高。一个故障是代码错误造成的,有时这种错误是不可避免的,但要修改的成本是很小的,因为这不是整体构架的错误。

1. **什么是软件工程？它有哪些本质特性？怎样用软件工程消除软件危机？**

1993年IEEE的定义：软件工程是：把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程，也就是把工程应用于软件；研究中提到的途径。

软件工程的本质特征：(1)软件工程关注于大型程序(软件系统)的构造(2)软件工程的中心课题是分解问题，控制复杂性(3)软件是经常变化的，开发过程中必须考虑软件将来可能的变化(4)开发软件的效率非常重要，因此，软件工程的一个重要课题就是，寻求开发与维护软件的更好更有效的方法和工具(5)和谐地合作是开发软件的关键(6)软件必须有效地支持它的用户(7)在软件工程领域中是由具有一种文化背景的人替具有另一种文化背景的人(完成一些工作)

消除软件危机的途径：(1)对计算机软件有一个正确的认识(软件≠程序）(2)必须充分认识到软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目(3)推广使用在实践中总结出来的开发软件的成功技术和方法(4)开发和使用更好的软件工具。

1. **简述结构化范型和面向对象范型的要点，并分析它们的优缺点。**

1.传统方法学：也称为生命周期方法学或结构化范型。

优点：把软件生命周期划分成基干个阶段，每个阶段的任务相对独立，而且比较简单，便于不同人员分工协作，从而降低了整个软件开发过程的困难程度。

缺点：当软件规模庞大时，或者对软件的需求是模糊的或会承受时间而变化的时候，开发出的软件往往不成功；而且维护起来仍然很困难。

2.面向对象方法学：优点：降低了软件产品的复杂性；提高了软件的可理解性；简化了软件的开发和维护工作；促进了软件重用。