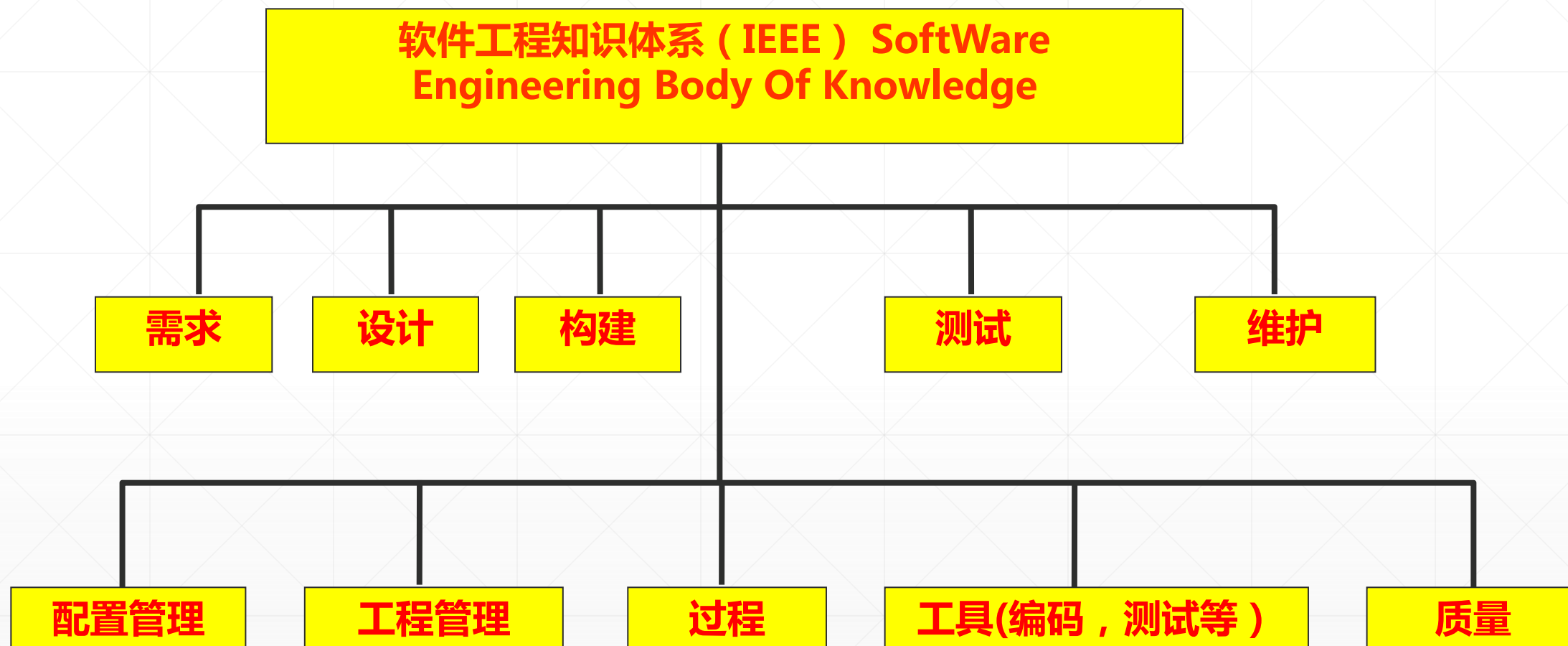
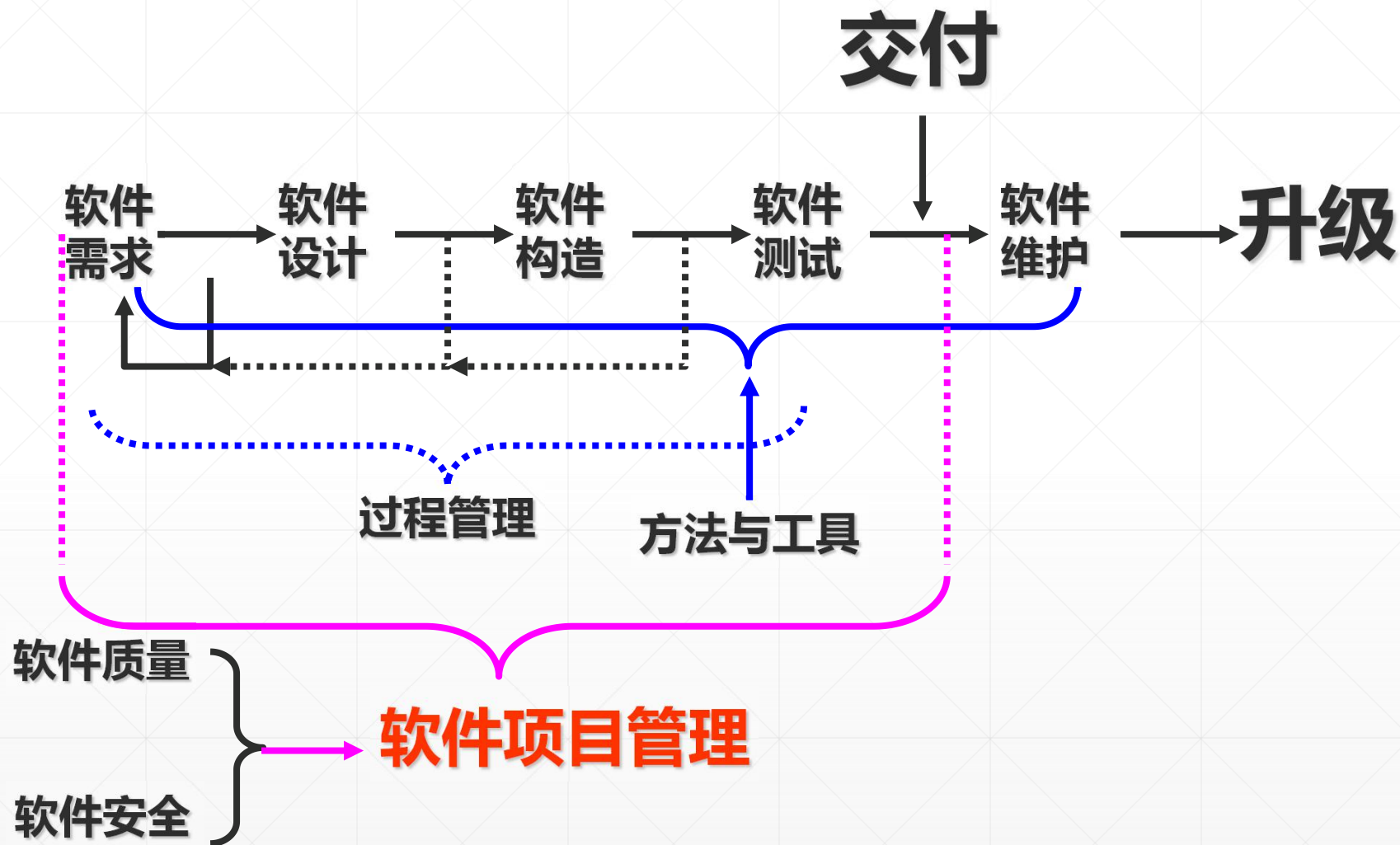




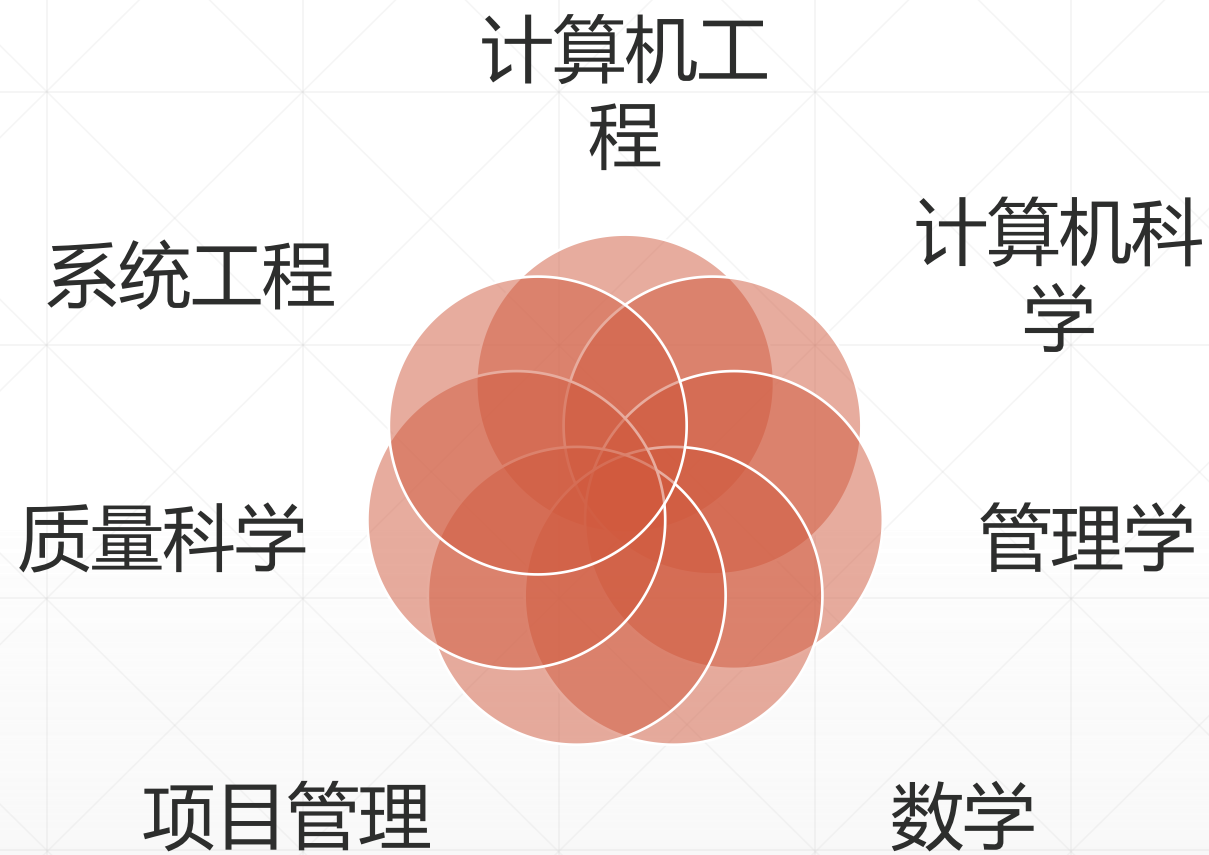
# 软件工程知识体系 (SWEBOK)



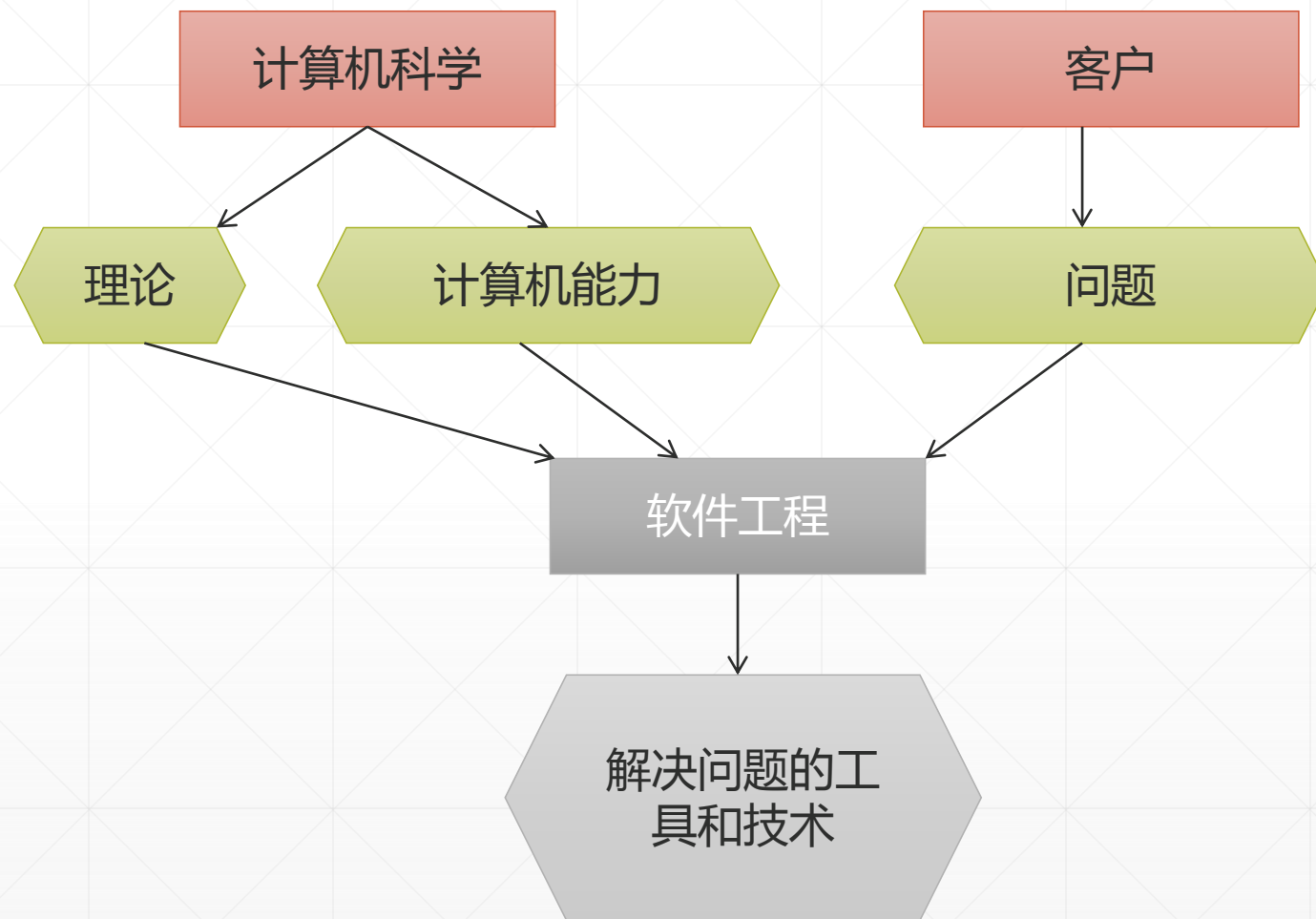
# 软件工程知识体系各主题之间的关联



# 软件工程是一门交叉学科



# 软件工程 VS 计算机科学



# 软件工程 VS 计算机科学

	软件工程	计算机科学
目标	在时间、资源、人员这3个主要限制条件下构建满足用户需求的软件系统	探索正确的计算和建模方法，从而改进计算方法本身
产品	软件（比如办公包和编译器）	算法（比如希尔排序法）和抽象的问题（比如哲学家进餐问题）
进度与时间表	软件项目都有特定的进度与时间表	研究项目一般不具有设置的进度与时间表
关注点	软件工程关注如何为用户实现价值	软件理论关注的是软件本身运行的原理，比如时间复杂度，空间复杂度，和算法的正确性
变化程度	随着技术和用户需求的不断变化，软件开发人员必须时刻调整自己的开发以适应当前的需求。同时软件工程本身也处于不断的发展中	对于某一种特定问题的正确解决方法将永远不会改变
需要的其他知识	相关领域的知识	数学

# 一些对软件工程的误解（1）

（注：M 代表误解，R 代表现实。）

管理方的误解：

M1: 我们已经有一本关于软件生产的标准和过程的书，这还不能让我们学习到需要的知识吗？

R1: 相比最新的大型主机，工作站和PC，这会使我们在做高质量软件开发时花费更多时间。

M2: 如果我们项目进度落后了，可以加入更多的程序员来赶进度。

R2: 软件开发的机制和手工作业不一样。在一个延迟了的软件项目中加入新的开发人员只会让它延迟更多。

M3: 如果我们将软件项目外包给第三方，我们就轻松了，让那个公司去完成它吧。

R3: 如果组织管理方不懂得如何从内部管理和控制软件项目，即使将项目外包也无济于事。

---

# 一些对软件工程的误解（2）

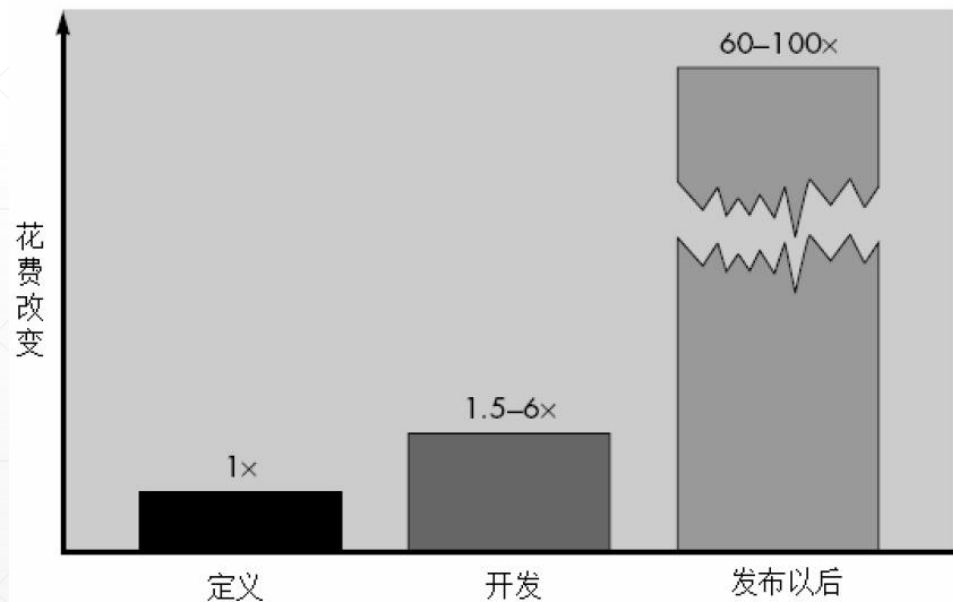
## 客户方的误解

M1: 对目标的一般陈述就足以开始编程，我们可以今后再补充细节。

R1: 前期糟糕的项目需求定义，是导致软件失败的主要原因。

R2: 项目需求的确在不断变化，但变化所产生的影响是根据变化提出的时间不同而不同的。

需求变化对变更成本的影响





# 一些对软件工程的误解（3）

开发人员: 软件从业者积累了50 年的编程文化。在早期的软件开发中，编程被视为一种艺术形式。

M1: 一旦我们编程完毕并成功运行，我们的工作就结束了。

R1: “越早开始写代码，我们就会花费越长的时间去完成它”。工业数据显示，软件开发60%-80%的精力将耗费在软件首次提交给用户以后。

M2: 当我的程序运行之前，我没有办法评估它的质量。

R2: 一个最有效的软件质量保证机制应当在项目的正式开始启动时——可以通过技术报告体现。

M3: 唯一可交付的工作成果是一个成功运行的项目程序。

R3: 一个可运行的程序只是软件结构的一部分，它还包含了许多其它因素。

M4: 软件工程将会让我们去创建大量不必要的文档，并且总是使我们的进度放慢。软件工程仅仅是文档而已。

R4: 软件工程并不是创建文档，而是创建质量。更好的质量减少返工的概率。更少返工会让项目更早交付。所有的文档都是提高团队沟通和质量所必须的。

# 软件工程职业道德和责任规范



1) 诚信：工程师们应当对他们的雇主和顾客时刻保持诚信而无论之前是否达成了关于诚信的协议。



2) 能力：工程师们不应该虚夸他们的能力水平。他们不应该故意接受一份超出自己能力范围的工作。



3) 知识产权：工程师们应该了解当地的知识产权法律法规，如专利权、版权等。他们应该小心确保雇主和客户的知识产权受到了保护。



4) 滥用计算机：软件工程师不以他们的工作职责为由滥用别人的电脑。滥用计算机的范围很广，从极小（在雇主的机器上玩游戏）到极其严重的（传播病毒）。



授课教师：蓝天      电子邮箱：[lantian1029@uestc.edu.cn](mailto:lantian1029@uestc.edu.cn)