1. 标题： 大道至简第七八章感悟
2. 发表时间：2015-11-10 19:00
3. 博客地址：<http://www.cnblogs.com/java-meng/p/4953968.html>
4. 内容：

现实中的软件工程，从最早仅仅关注于软件开发工具到现在，软件行业中的巨头们已经在层出不穷的思想中涅槃了一回又一回。大公司们在标准，理论，语言上的争来争去，未必全然出于“软件实现”的考虑。对统一理论，统一工具，统一过程的企图，其最终目的是在整个软件工程体系中的全胜而出。因而，除了软件本质力量的推动之外，商业因素也推动着软件工程体系的发展。大公司们的争夺战的最终结果，已经开始把软件工程，从原始的“自生演进”状态，逐渐推进到“它激发展”的状态上了。这种它激发展可能会影响到软件工程发展的速度，然而在各个工程层面上的关注点并不会发生变化。从角色的角度上来说：开发经理思考项目的实施方案和管理具体的开发行为；而项目经理则保障团队的稳定性和一致性。然而这只是基本模式，或者说，是理想模式。

      理想状况下，“软件工程=过程+方法+工具”。然而工程成功的真正关键，并不是在于你把你的团队“组织”得非常好。即使在团队中他们都显示有条不紊，你一样会面临失败。正如前面所说，如果你是一个软件公司里的项目经理，你可能今天的工作是写一份项目计划案，或者听测试部的报告，又或者是安排会议来听取和分析一个新的产品需求。

      过程伴随工程而出现，工程又是如何出现的呢？根本的原因是软件规模的不断增大所导致的。随着软件规模的的增大，仅仅一个人的话花费的时间是巨大的，在现实中不会有软件公司给这样的机会的。项目的“复杂”可能可能需要不同知识领域的角色参与，而“庞大”则要求更多的资源。“团队”作为开发行为的模式，是软件规模和复杂度渐次累积的结果。团队越来越庞大，因为软件规模越来越复杂。没有团队意识的软件公司将在高度过程化，通晓方法理论，拥有大量工具的集团军面前一触即溃。

     思考问题的方法可以是由点及面的，也可以是统揽全局的。换成业界最常用的词汇，就是“自上而下”还是“自下而上”的区别。RUP是对前人在软件过程思想方面的高度包容。它如同一个杂货箱一样放满了各种稀奇古怪的东西。RUP能不能被用起来，将取决于你挑挑拣拣的行为。出于共同的必要，UML的象征意义在一个图中应当被表述得足够准确和详细，乃至于针对于不同的阅读者来说都能提供了充足的信息。所以在工程中使用UML图，应该有相应的文字来描述它。而且这种描述与图之间的对应关系要持续的维护下去。所以UML有了属于它自己的规约。