敏捷(Agile)作为一种开发流程， 目前为各大公司所采用, 敏捷流程的具体实践有XP 和Scrum, 似乎很少有文章介绍这两者的区别,

发现一篇外文, 见解非常深刻, 特将其翻译一把.

原文(DIFFERENCES BETWEEN SCRUM AND EXTREME PROGRAMMING )在此:

http://blog.mountaingoatsoftware.com/differences-between-scrum-and-extreme-programming

作者总结的大致区别如下：

区别之一：迭代长度的不同

XP的一个Sprint的迭代长度大致为1~2周, 而Scrum的迭代长度一般为 2~ 4周.

区别之二: 在迭代中, 是否允许修改需求

XP在一个迭代中，如果一个User Story(用户素材, 也就是一个需求)还没有实现， 则可以考虑用另外的需求将其替换， 替换的原则是需求实现的时间量是相等的。 而Scrum是不允许这样做的，一旦迭代开工会完毕, 任何需求都不允许添加进来，并有Scrum Master严格把关，不允许开发团队收到干扰

区别之三: 在迭代中，User Story是否严格按照优先级别来实现

XP是务必要遵守优先级别的。 但Scrum在这点做得很灵活， 可以不按照优先级别来做，Scrum这样处理的理由是： 如果优先问题的解决者，由于其它事情耽搁，不能认领任务，那么整个进度就耽误了。 另外一个原因是，如果按优先级排序的User Story #6和#10，虽然#6优先级高，但是如果#6的实现要依赖于#10，则不得不优先做#10.

区别之四：软件的实施过程中，是否采用严格的工程方法，保证进度或者质量

Scrum没有对软件的整个实施过程开出严格工程实践的处方。要求开发者自觉保证，但XP对整个流程方法定义非常严格，规定需要采用TDD, 自动测试，结对编程，简单设计，重构等约束团队的行为。因此，原作者认为， 这点上，XP的做法值得认同的，但是却把敏捷带入了一个让人困惑的矛盾, 因为XP的理念，结合敏捷模式，表达给团队的信息是“你是一个完全自我管理的组织， 但你必须要实现TDD, 结对编程，...等等”

不难发现，这四个区别显见的是： Scrum非常突出Self-Orgnization, XP注重强有力的工程实践约束

作者建议， 在管理模式上启用Scrum， 而在实践中，创造一个适合自己项目组的XP(“start with Scrum and then invent your own version of XP.”)