关于RUP方法的感悟

在了解RUP方法时，时常有一些感悟，所以本篇感悟是在了解的过程中边读边写的，并非了解完之后一蹴而就。根据定义来说，RUP（Rational Unified Process）是一个面向对象且基于网络的程序开发方法论。通俗理解，就是一套指南，其中包括了整个过程的各个阶段以及所用的组件工具。

关于“六大经验”。

1. 迭代式开发。在初期时开发人员是不可能完全非常清晰了解到用户需求以及所要开发的项目的各个细枝末节的，一方面是用户无法完全正确表达他的需求，另一方面开发人员无法准确理解，所以在整个开发过程中，我们更像是画一棵树一样，先画出主干，再去画枝干与叶子，而在这个项目工程中，我们所有的文档都需要版本号来记录每一次迭代过程，这也是为什么我们要用git来管理。
2. 管理需求。整个项目工程中，代码开发和测试所用的时间并不多，大部分的时间用于前期的各种准备阶段，为了防止在后期出现问题而导致付出过高的成本，所以在代码开发之前需要将需求等文档化，以文字来描述。
3. 基于组件的体系结构。软件重用能够提高软件生产率，降低开发成本，缩短开发周期，且有助于改善软件质量，。
4. 可视化建模。使用UML，可视化模型的建立，有效管理软件复杂性。
5. 验证软件质量。在传统开发中，质量评估往往是在后期的活动，在RUP中，这项活动建立在整个过程中，避免出现过晚发现缺陷而付出高代价。
6. 控制软件变更。为了防止开发人员在开发过程中混乱的迭代，而需要给每个开发人员建立独立的空间。

RUP的软件生命周期被分解为了四个阶段，按顺序是：初始阶段、细化阶段、构造阶段以及交付阶段。每个阶段结束后有一个里程碑，标志阶段的结束和对这一阶段结果的评估。

1. 初始阶段。定义上说，初始阶段的目标是为系统建立商业案例并确定项目的边界。为了达到该目的必须识别所有与系统交互的外部实体，在较高层次上定义交互的特性。这是一个短阶段，在我认为，我们目前所作的项目介绍中定义好产品功能、用户等是包括在这个阶段内的。这个阶段更多的关注于目进行中的业务和需求方面的主要风险。
2. 细化阶段。细化阶段的目标是分析问题领域，建立健全的体系结构基础，编制项目计划，淘汰项目中最高风险的元素。在这一阶段，我们细化功能需求，性能要求，定义环境与使用工具等，我们目前做的项目计划以及需求分析应是属于这一阶段的产物，也是这一阶段结束时用来评估的里程碑。
3. 构造阶段。通俗理解就是代码开发+测试，是产品的制造过程，要确定软件、环境、用户是否可以开始系统的运作。
4. 交付阶段。也就是经过最后少量的测试调整，完成工程，将产品发布在应用商店。

除此之外，RUP还有9个核心工作流：6个核心过程工作流和3个核心支持工作流。工作流在项目中轮流被使用，在每一次迭代中以不同的重点和强度重复。

1. 商业建模。关于这一点不是很懂，个人理解为，这个目标在社会或对象群体中起到什么作用？
2. 需求。
3. 分析和设计。这一部分应该包括了可行性分析、需求分析，在后面的阶段中，需要经常那个用到这两份分析报告，并且会根据他们设计出抽象模型，将整个产品以结构视图的方法直观呈现。
4. 实现。即代码开发，很重要的一点，基于组件的体系结构，尽可能多的实现软件重用。
5. 测试。RUP方法将测试置于整个过程中，而非实现部分完成之后，尽可能早地发现缺陷，从根本上降低了修改缺陷的成本。
6. 部署。
7. 配置和变更管理。因为RUP方法采用迭代式，所以在文档、产品的每次更新迭代，都需要阐述对产品修改原因、时间、人员保持审计记录。
8. 项目管理。包括为项目的管理提供框架，为计划、人员配备、执行和监控项目提供实用的准则，为管理风险提供框架等。
9. 环境。

RUP的每个阶段可进一步分解为一个迭代过程，也就是说，阶段到阶段，是一个迭代过程到另一个迭代过程，每一个迭代过程是一个循环，也就是工作流的一次顺序完成。通过一次次循环经过，减少传统方法中在分析、设计和实现阶段遗留的隐藏问题的情况，减少解决问题而付出的成本。总的来说，RUP方法比传统方法相比，最大的优点就是减少了成本，包括开支和时间，另外，RUP的迭代过程更加灵活，在需求发生变化时，能快速调整以适应变化。