软件构建管理

软件构建管理：

（1）版本控制

版本控制功能执行诸如工作空间创建和更新，基线和报告的活动。它为创建过程创建了一个运行环境，并捕获有关构建过程的输入和输出的元数据，以确保可重复性和可靠性。

诸如Git，AccuRev或StarTeam等工具通过提供工具将历史中的特定点标记为重要和更多来帮助这些任务。

版本控制功能执行诸如工作空间创建和更新，基线和报告的活动。它为创建过程创建了一个运行环境，并捕获有关构建过程的输入和输出的元数据，以确保可重复性和可靠性。

诸如Git，AccuRev或StarTeam等工具通过提供工具将历史中的特定点标记为重要和更多来帮助这些任务。

这次项目配置管理活动，我们选择的是使用Git作为版本控制工具，由于Git的分布式特点，极大程度上简化了团队之间的协作，对于多人同时进行的项目构建有非常大的帮助。

（2）代码质量

这个部分需要负责检查开发人员是否坚持了保证代码质量的七个原则：注释，单元测试，代码是否重复，代码的复杂性，编码规则，潜在的Bug和架构的设计。确保项目具有高质量的代码，可减少错误，并影响非功能性需求，如可维护性，可扩展性和可读性，这对业务的投资回报率有直接影响。

（3）编译

这只是管理构建过程的一个小功能。编译函数将源文件转换为可直接执行的或中间对象。不是每个项目都需要这个功能。而对于简单程序，该过程由正在编译的单个文件组成，对于复杂软件，源代码可以包括许多文件，并且可以以不同的方式组合以产生许多不同的版本。

关于软件构建管理工具：构建计算机程序的过程通常由构建工具来管理，构建工具是协调和控制其他程序的程序。这样的程序的例子是make，Gradle，Meister by OpenMake Software，Ant，Maven，SCons和Phing。构建实用程序通常需要以正确的顺序编译各种文件。如果特定文件中的源代码没有改变，则它可能不需要被重新编译（可能不是因为它本身可能依赖于已经改变的其他文件而不需要重新编译）。复杂的构建实用程序和链接器试图避免重新编译不需要它的代码，以缩短完成构建所需的时间。更复杂的过程可能涉及其他程序产生代码或数据作为构建过程的一部分