权限控制，fancyUI 等）；源代码文件夹存放与本项目相关的数据库在内的所有源代码，本项目采用 github 的方式管理代码，对每个版本的代码都打上了 tag 标签并作记录，方便相关代码随时 checkout 出来；可执行文件夹存放项目经过部署后的相关可执行文件及运行环境所需要的第三方安装包，如 tomcat 服务器，Hadoopjar 等文件；测试用例文件夹存放各功能模块的冒烟测试用例，集成测试用例，系统测试用例等测试用例文件。测试报告文件夹存放与测试报告相关的各类文件，包含每个版本修正的问题，该版本数据库，算法分析，页面展示的性能报告，压力测试报告等与测试报告相关的所有文档。使用手册文件夹存放系统介绍说明书，系统使用最佳实践，与竞品比较等其他说明性文档。同时定义了版本的基线包含设计文档文件夹与源代码文件夹，非基线配置包括其他的各类计划，报告等文档。对于基线配置，项目组所有研发人员有读权限，对于非基线配置，配置变更委员会的成员有读写权限。

三、配置控制

配置控制是控制配置变更的过程，该过程的实施对于项目开发至关重要。因为随着项目的深入实施，配置项的新增或修改是无法避免且可能是高频的，所以变更控制非常重要，实施不好，代码库就会产生歧义，无法得到当初客户想要的版本，也可能无法描述清楚该版本到底有什么新特性对应的测试报告在哪里等问题。所以我作为项目经理，对该过程的实施非常重视。

在本项目中，我严格要求对于配置项的变更需要提出变更申请，例如在一次代码评审会上，测试员提出经过测试发现当数据中包含时间类型数据量超过 500 万后，经 CNN 神经网络算法分析前端页面白屏，初步测试人眼分析发现后端接口超时，不满足企业方要求的在 5 秒内出结果并返回到前端的要求，后经过开发人员与算法架构师讨论，算法确实存在性能问题，需要优化，于是提出了软件版本的升级，希望解决这个问题，由于这个问题修改的面比较广，涉及数据库的并发读取优化与算法优化，传输数据优化等内容，所以我提出修改后将软件版本从目前的 1.0 升级到 1.1 版本。我组织配置变更委员会 CCB 成员开会讨论，在充分考虑本次修改影响的业务模块，影响的效果及优化方案是否可行后，一致同意批准其修改。我们将同意修改的结论告诉了项目组全体成员，之后项目组开发人员大数据开发组与算法组协同前端传输开发开始实施修改，优化了代码，经过测试组测试验证并确认系统没有再出现接口超时的现象，之后将主版本号变更并发布，最后项目组开发人员实施基于配置库的变更控制，首先将 V1.0 代码从产品库中取出 copy 放入受控库，由于本修改涉及多个开发组多个人，开发人员从受控库 checkoutV1.0 代码到开发库中准备修改，此时代码版本为 V1.01，此时受控库代码为 lock 状态，其他开发人员不能再 checkout，等待代码 unlock 后才能 checkout 最新代码，大数据开发人员修改完毕后，checkin 代码，此时受控库代码改为 unlock，算法组 checkout 受控组最新代码，此时受控库代码再次被 lock 住，当算法组 checkin 代码后，受控库改为 unlock 状态，build 工程师根据受控库 build 出最新可执行文件，经过测试组测试无误后，将受控库的代码 update 存入新的基线版本中，并将版本号从 V1.0 改为 V1.1，同时保留 V1.0 的代码及标签。

四、配置状态报告

配置状态报告主要是检查各配置项当前的状态以便采取适当的措施使其完整管理本项目各配置项的状态。我作为项目经理，随时关注了每个配置项当前的状态和已批准的实施状态，并比较之前版本与现在版本配置项的变化情况，对于没有处理完的配置项需要特别关注实施进度。

我每周都会抽出固定时间关注每个配置项的状态并和配置管理员开会确认目前配置项的状态监控结果。当开发组经理提出配置项变更申请时，配置管理员将配置项改为分析状态；经过配置变更委员会评审后，变为审批状态；再经过项目组实现后，变为验证状态。并记录配置项变更的每个记录。

五、配置审计