

应式的 Web 设计和移动优先的策略, 确保用户在不同设备上都能获得良好的使用体验。

软件配置管理是保证软件质量的关键, 也是保证软件产品交付的重要手段。有效的项目配置管理, 能够保证软件工作产品在整个项目周期中的“完整性”, 提高软件的工作效率, 及早发现风险并制定相应风险应对措施, 提升软件项目管理水平, 同时也能增强用户和公司对于软件质量和项目组的信心。本《智慧城市平台》项目工作量大, 涉及 10 个大的业务子系统与 30 多个功能模块, 产生的设计文件、说明文档、各种原型图、类图、源代码等非常之多, 各业务子系统采用敏捷式开发, 版本迭代频繁, 如果不能对这些资料、文档进行有效管理, 保证势必引起混乱, 因此做好项目配置管理尤其重要。下面我将结合本次项目实践叙述项目配置管理的具体做法。

### 一、制定配置管理计划

制定配置管理计划是规划如何开展项目配置管理工作, 是配置管理过程的基础。项目之初, 我邀请项目顾问, 各技术组长及项目骨干, 专题会议制定了《智慧城市平台配置管理计划》, 主要内容有: 一是确定软件配置管理所使用的工具、技术和方法, 使用 KCFLOW 配置管理平台, 实时自动化约束并客观记录配置管理活动。二是确定配置管理机构 and 人员职责, 成立了由甲方信息主任周工、客户代表、我、项目技术负责人、配置管理员、QA 等组成的软件配置控制委员会, 负责对变更进行评审批准, 并确认变更的结果; 配置管理员负责制定《配置管理计划》并执行; QA 负责审计配置管理过程; 开发组组长, 负责评估变更中代码方面的问题。三是确定了变更申请流程、变更申请方法, 以及对配置管理报告内容、报告时机、报告人和通告对象等进行了描述。该配置管理计划经过 CCB 审批后, 成为配置管理活动的指南和方向。

### 二、配置标识

配置标识是识别需要受控的配置项并记录配置项的功能和物理特征。在该《智慧城市平台》项目中, 配置管理员、QA、技术骨干等利用 KCFLOW 配置管理平台, 建立了功能基线、分配基线和产品基线 3 条控制基线。识别出了需要受控的配置项, 有需求相关文档、设计相关文档、测试文档、编译程序、程序代码等, 并为每个配置项指定了唯一性的标识号, 版本标识规则统一为: vm.n.j (其中 m 代表主版本号, n 代表次版本号, j 代表批准次数)。确定了每个配置项的所有者及其责任, 为配置管理工作奠定了基础。

### 三、配置控制

配置控制是对配置项和基线进行变更控制。对已进入受控库和产品库的任一软件配置管理项的更改, 都需履行申请和审批手续。如项目中期, 城市事业指导科的李科长提出要添加疫情防控视频信号定时轮巡的功能, 我们要求李科长填写书面《变更申请单》, 然后提交给 CCB。CCB 对变更申请评估后, 批准了该申请, 并将决定通告每个干系人。我们项目组变更实施人填写《变更出库单》, 审批通过后, 从受控库中检出 (check out) 待更改的配置项到开发库中进行修改, 则该配置项处于待“锁定”状态, 以保证同一配置项只能同时被一个变更实施人员修改。变更完成并通过验证后, 变更实施人填写《变更入库申请单》, 审批通过后, 将变更后的配置项检入 (check in) 受控库中, “锁定”解除。最后, 软件产品修改工作全部完成后, 更新至产品库中, 并更新了产品版本号。

### 四、配置状态报告

配置状态报告的任务是有效地记录和报告配置项的当前状况, 供相关人员了解, 以加强配置管理工作。比如, 对“城市治理疫情防控”模块, 刚开始创建时处于“草稿”状态、版本号为 0.01, 后经过多次完善, 版本号升级为 0.12; 经过批准后, 该配置项成为“正式”状态、版本号为 1.1; 后来, 李科长提出要添加疫情防控视频信号定时轮巡的功能, 则需对该配置项进行更改, 此时该状态变为“修改”、版本号为 1.11。再次经过评审后, 其状态变为“正式”, 版本号为 1.2。在整个过程中, 配置项不停地演化, 配置管理员时刻观察每个配置状态, 记录和保存每个变化的版本和状态, 以便更好地进行控制。