

目标拆解与过程管理

开源组件库并进行多端适配

Specific (具体)

- 在1个月内开源项目的初步版本，开源的同时进行脱敏关键信息，去掉机密信息
- 将现有的小程序组件库改造成同时支持小程序和Web H5的组件库，1个月内完成
- 将项目改造成monorepo架构，拆分成多个软件包，并发布到npm上，1个月内完成
 - 小程序组件软件包
 - 通用组件软件包
 - 演示Demo软件包
- 确保新提交的代码质量和风格符合标准，1天内进行Code Review
- 编写完善的文档，随着项目的更新及时维护

Measurable (可衡量)

- 组件库在小程序和Web H5上的使用效果经过内部测试确认
- 软件包成功发布到npm上，并且可以正常引入
- 公司内部信息已经被脱敏，开源项目已经发布到公共代码仓库
- 新提交的代码经过Code Review，符合标准
- 文档编写并及时维护，反映组件库的最新状态

Achievable (可实现)

- 基于现有技术栈和开源社区，确保能够改造出同时支持小程序和Web H5的组件库
- 软件包拆分和发布的工作可以在现有团队资源和时间内完成
- 项目开源前能够去掉机密信息，并且具备一定的代码质量和风格标准
- Code Review流程能够被嵌入到现有开发流程中，并且能够被团队成员接受和执行
- 文档的编写和维护可以在现有团队资源和时间内完成

Relevant (相关性)

- 该项目的改造和开源符合公司和团队长期发展战略和价值观

- 该项目的改造和开源能够为公司和开发者社区提供有价值的组件库资源
- 该项目的改造和开源能够提高团队的技术水平和职业发展机会

Time-bound (时间限定)

- 目标在限定时间阶段内符合SMART原则，以时间维度拆分目标
- 季度维度
 - 开源组件库并进行多端适配
- 月维度
 - 在1个月内完成项目的开源，并且在开源的1个月内去掉机密信息
 - 将组件库改造成同时支持小程序和Web H5的组件库时间限定为1个月
 - 将项目改造成monorepo架构、拆分成3个软件包、并发布到npm上的时间限定为1个月
 - 确保新提交的代码在1天内进行Code Review
- 周维度
 - 第1周
 - 完成技术方案的调研和规划，确定技术方案，检测技术方案的可行性
 - 第2周
 - 新建私有开源代码仓库
 - 对现有项目进行调整，去除机密信息和内部包
 - 对组件库进行移植，将项目迁移到公有代码仓库，并测试
 - 配置github workflow，确保CI/CD能够正常使用
 - 进行发布到npm上的准备工作
 - 根据项目开源进行文档的改造和补充
 - 第3周
 - 对软件包进行本地测试和发布npm测试
 - 私有代码仓库开源zaihui组织
 - 将现有项目进行monorepo架构的改造环境改造、子包项目搭建
 - 进行项目和依赖剥离，将组件库拆分成组件和demo两部分，并分别移植到小程序子包和Demo子包内
 - Demo小程序能正常启动，测试分包后组件库组件能够在Demo内正常展示，能够在Demo小程序中正常运行
 - 第4周
 - 进行组件库的测试，如样式、交互等，确保在小小程序组件库开源后能正常运行

- 对组件库代码进行重构和优化，提高组件库的可重用性和可维护性
- 配置monorepo架构下的github workflow
- 进行monorepo架构下包发布到npm上的准备工作
- 第5周
 - 对软件包进行本地测试和发布npm测试
 - 配置软件包的CI/CD, workflow接入公司内部的服务服务器
- 第6周
 - 对软件包进行正式版本npm发布，确保发布的包能够正常使用
- 第7-8周
 - 小程序组件改造h5，对组件库进行拆分和整理，梳理出能够做h5通用适配的小程序组件
 - 新增一个shared包，拆分成小程序组件包和通用组件包和Demo三个独立的软件包
 - 进行组件库移植和改造，逐一测试组件的兼容性和可用性，保证Web H5使用时的稳定性和体验
- 第9周
 - 进行后续的完善文档工作，包括组件库的使用说明、API文档、示例代码等
 - 对项目进行最后一次整体测试和代码检查，确保发布的软件包质量符合要求

PDCA原则

计划（Plan）：

1. 确定项目的目标：在2023年第二季度完成小程序组件库的开源及H5适配
2. 制定详细的计划和时间表：确定需要开源的范围和功能开发的耗时，制定一个具体的时间表来保证项目按时完成以及每个部分需要花费的时间和资源

执行（Do）：

1. 小程序组件库开源，能够正常进行功能迭代及开源维护
2. 对小程序和h5环境同时兼容适配

检查（Check）：

1. 检查项目的进展情况：根据时间表，检查项目的进展情况并及时跟进
2. 检查项目的质量：对每个合入开源组件库的开源PR进行Review，确保开源库的稳定迭代和统一的规范风格，保证合入的代码符合组件库的规范

优化（Act）：

1.

根据检查结果，对需要改进的地方进行优化
2.

确定是否目标都已实现，没有达到的将进行目标的重新评估