# KOCA-UI扩展组织管理规范

**依赖现状:**

****

图 1KOCA-UI依赖现状

**规范要点：**

**①KOCA-UI如何有效并入“中心团队”的扩展组件**

**②KOCA-UI并入“中心团队”的扩展组件后对业务项目不会造成太大影响，业务团队的项目升级平滑**

## 一、KOCA-UI如何并入“中心团队”的扩展组件？

****

子模块

[https://gitlab.com/CF/kace-ui.git](https://gitlab.com/xxx/kace-ui.git)

### git submodule 管理各子模块的组件

**注：g**it submodule的功能是建立了当前项目与子模块之间的依赖关系：子模块路径、子模块的远程仓库、子模块的版本号。https://zhuanlan.zhihu.com/p/87053283

**具体实施步骤：**

1. 业务部门维护并管理好业务组件仓库：[https://gitlab.com/CF/kace-ui.git](https://gitlab.com/xxx/kace-ui.git) (维护标准)

 2. KOCA-UI 添加子模块组件库:

git submodule add -b [kace-ui] https://gitlab.com/CF/kace-ui.git componets/kace-ui

### 业务组件库发布管理：

 **发布方式:**

分布式发布

 szkingdom.yf.koca-ui

@szkingdom.yf.koca-ui/kace

@szkingdom.yf.koca-ui/fs2.0

优点：koca-ui和各子模块业务组件独立运维和管理



 **发布流程：**

****

 KOCA主导发正式版，发版前KOCA测试检查业务组件是否符合开发规范（布局，颜色等是否做好了兼容），统一视觉，统一交互；

业务方开发使用发snapshot版

需要缩短（测试，验收）时间，快速响应

## 二、KOCA-UI合并“中心团队”扩展组件后，项目升级KOCA-UI后对项目无影响

### Koca 提供业务组件库基础模板及规范

**业务组件库模板:**

* 样式预编译语言： sass
* Js: ts/es6
* 框架：vue



**组件库原则**：

“约定大于配置”

1. packages文件夹存放组件相关代码
2. package.json组件包的描述信息
3. packages/assets文件夹存放公共样式（可选）
4. packages/assets/fonts 存放图标文件（建议尽量复用koca-ui提供的图标）
5. packages/assets/scss 存放公共样式代码（一般不会用到）
6. 组件放在packages文件夹下
7. 组件命名：小写字母+短横线
8. Packages/index.js 组件统一引入
9. 组件在开发时不加前缀，注册时统一加前缀
10. 支持默认引入、按需引入
11. 组件开发时需考虑到主题切换问题，对颜色布局进行抽象