

前端组件库管理规范

规范目的

- 提高开发人员开发效率
- 避免组件重复编写，造成存在多个相同或相似组件
- 便于后期维护
- 便于输出规范文档

代码规范

代码规范应遵循 [《前端代码规范\(PC端Vue2.x\)》](#)

组件库管理要求

1. 组件库代码应存放在公司指定代码管理库，并由专人负责，除了管理人员，开发人员无权对组件库进行修改。
2. 组件库通过测试后提交至npm库，在项目中通过npm进行引用。
3. 组件库需要更新时除了已知bug，如需对组件进行修改、更新应需要产品经理、项目经理发送修改邮件，管理人员针对修改邮件做出评估意见。
4. 组件版本封存，通过单元测试、测试同事测试通过后应对版本进行封存，并上传至npm库。
5. 组件库版本管理应该遵循x.x.x(版本号.次版本号.修订号)
 - 版本号：对组件新增/下架需要迭代+1
 - 次版本号： 针对一个/多个组件业务逻辑进行调整需要迭代+1
 - 修订号：已知存在的bug修改迭代+1。如修订号达到10，次版本号应+1
6. 代码库应由专人管理，所有组件应生成相应文档，交接时应对组件库、文档进行交接。
7. 管理人员应对组件库组件逻辑具有深入的理解，对组件业务逻辑具有清晰的了解。
8. 组件变更应在最新版本进行更新，严禁在已封存版本中进行修改。

代码封装原则

1. 单一原则：负责单一的页面渲染
2. 多重职责：负责多重职责，获取数据，复用逻辑，页面渲染等
3. 明确接受参数：必选，非必选，参数应避免变量重复
4. 可扩展：需求变动能够及时调整，不影响之前代码
5. 代码逻辑清晰
6. 封装的组件必须具有高性能，低耦合的特性
7. 组件函数如果为内部使用，应使用私有函数，不应向外暴露。
8. 组件具有单一职责：封装业务组件或者基础组件，如果不能给这个组件起一个有意义的名字，证明这个组件承担的的职责可能不够单一，需要继续抽组件，直到它可以是一个独立的组件即可。

组件修改原则

1. 组件修改应由专人维护，如果需要更改组件库应产品经理发送组件修改、组件新增邮件，邮件中应说明需要改动的内容。组件管理人员根据需求结合管理原则作出评估。

2. 组件封装完毕经单元测试、测试同事、项目验收通过后，除了修复存在的bug，原则上不支持修改组件代码。
3. 对组件进行扩展，应对组件进行评估，确保修改后组件不会对之前代码造成影响，如果会造成影响，对之前代码改动应降至最低。
4. 应结合当前组件职责和新需求进行分析，是否在原业务基础上进行修改，如果不是则应独立开发新的组件。

组件新增原则

- 1、需要新增组件，应确保组件库中没有具有相同组件职责的组件，且该组件在组件库中是唯一的。
- 2、新增组件完成后通过测试达到引入项目后，应形成文档并对开发人员进行培训。
- 3、新增组件应对组件进行三个使用版本以上的使用追踪，及时修改存在的bug。

组件退出原则

1. 经与产品经理、项目经理确认后续该组件不再使用，可进行准备退出，需要经过三个使用版本观察后确认对项目无影响后才能从组件库退出。
2. 该组件已经无法满足当前业务需求，并有了新的组件替换，需要经过三个使用版本观察后确认对项目无影响后才能从组件库退出。
3. 针对退出组件具有突破性技术的，应对该技术进行技术备份后才能进行清退。
4. 清退只能在最新版本中进行清退，旧版本应保留(如2.0.0版本清退该组件，2.0.0以下的应保留，2.0.0以上的版本不再支持该组件使用)