**软件版本控制文档**

**软件版本控制文档是一份记录软件版本控制的文档，其中包含了软件版本的历史记录、变更内容、开发计划等信息，可以帮助团队成员更好地协作开发和维护软件。**

**版本号：**软件版本控制文档中包含了每个版本的版本号，一般使用x.y.z的格式表示，其中x表示主版本号，y表示次版本号，z表示修订版本号。

**版本历史记录：**软件版本控制文档中包含了软件的版本历史记录，记录了每个版本的发布时间、变更内容、问题修复情况等信息。

**变更记录：**软件版本控制文档中包含了每个版本的变更记录，记录了对软件进行的每个更改，包括新功能、功能修改、Bug修复等。

**开发计划：**软件版本控制文档中包含了开发计划，记录了每个版本的开发计划、发布时间和目标。

**分支管理：**软件版本控制文档中记录了分支管理策略，例如主分支、开发分支、发布分支等，以及分支的合并和处理方式。

**发布管理：**软件版本控制文档中包含了发布管理策略，记录了如何进行版本发布、版本发布的流程和规范等。

**团队协作：**软件版本控制文档中包含了团队协作策略，例如如何协作、如何管理分支、如何处理冲突等。

**版本号**

当前版本号：v1.0.0

**版本历史记录**

**v1.0.0 – 20XX/01/01**

* 首次发布

**变更记录**

**v1.0.0**

* 添加了XX功能

**开发计划**

**v1.1.0 – 20XX/02/01**

* 添加了XX
* 修复了XX
* 优化了性能

**分支管理**

* 主分支（master）：用于发布稳定版本，不允许直接修改
* 开发分支（dev）：用于日常开发，可以进行功能开发和Bug修复
* 特性分支（feature）：用于开发新功能，从开发分支创建，开发完成后合并回开发分支
* 发布分支（release）：用于准备发布版本，从开发分支创建，进行Bug修复和版本测试，测试通过后合并到主分支和开发分支

**发布管理**

* 版本号规则：主版本号.次版本号.修订版本号
* 发布流程：开发完成后，从开发分支创建发布分支，进行Bug修复和版本测试，测试通过后合并到主分支和开发分支，并打上新版本号的tag

**团队协作**

* 协作方式：使用SVN进行代码托管和版本控制，遵循代码审查、单元测试、集成测试等规范
* 分支管理：从开发分支创建特性分支，开发完成后合并回开发分支，从开发分支创建发布分支进行版本发布
* 冲突处理：在分支合并时，如果有冲突，需要进行手动合并，遵循“合并后能够编译通过且测试通过”原则