

软件开发规范v1.1.0

时间：2023-04-17

版本：v1.10

修订人：

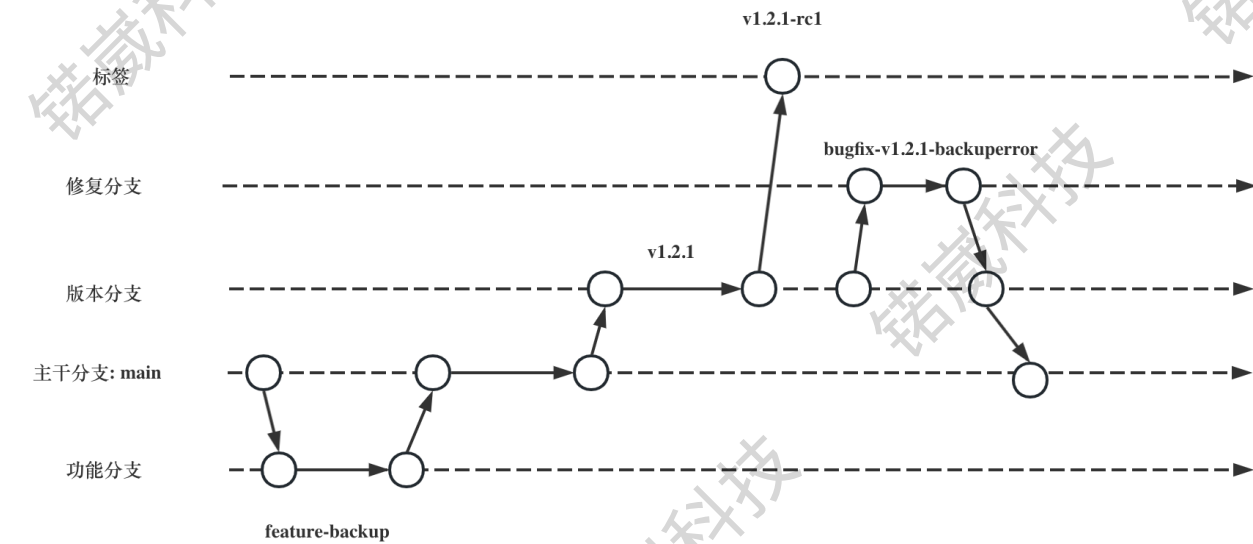
	A	B	C
1	修改时间	修改人	修改内容
2	2023/4/14 16:00	胡正兴	增加版本控制

软件开发规范

本文旨在规范公司软件开发流程，以确保高质量、高效率的软件开发。以下是具体规范要求：

一、版本控制

1. 开发流程



- 功能开发：开发人员从主分支main拉取本地功能分支进行开发，开发自测完成后提交PR，由负责人合并到main上；
- 版本提测：负责人基于main拉取版本分支v1.2.1，再基于v1.2.1打tag：v1.2.1-rc1，提交测试；
- bug修复：开发人员基于版本分支拉bugfix分支，修复完成之后提交PR，负责人合并到版本分支。版本分支bug修复验证测试完毕后再合并到main上。

2. 分支规范

	A	B	C
1	分支类型	命名规则	说明
2	主干分支	main	开发人员拉取本地分支进行开发
3	版本分支	v<主版本号>.<子版本号>.<修订版本号>	提交测试时，基于主干分支拉一个版本分支，例如：v1.1.1。
4	功能分支	feature-*	开发人员基于主干分支，拉取本地开发分支，例如：feature-datatamigration
5	修复分支	bugfix-<版本分支>-<BugID>-*	基于版本拉取修复分支，例如：bugfix-v1.1.1-196-删除迁移敏感信息日志

1. 主干分支：命名为 **main**，用于开发人员拉取本地分支进行开发；
2. 版本分支：命名为 **v<主版本号>.<子版本号>.<修订版本号>**，提交测试时，基于主干分支拉一个版本分支，例如：v1.1.1。
3. 功能分支：命名为 **feature-***，开发人员基于主干分支，拉取本地开发分支；例如：feature-datatamigration；
4. 修复分支：命名为 **bugfix-<版本分支>-<bugID>-***。基于版本拉取修复分支，例如：bugfix-v1.1.1-196-删除迁移敏感信息日志；

3. Commit规范

Commit Message格式：<type>:<subject>

type说明

以下是commit时可以使用的type：

- feat：新增功能，例如添加新的页面或者新的功能模块；
- fix：修复错误，例如修复程序中的bug；
- chore：对构建系统或配置文件、脚本的更改，例如升级maven或npm；
- docs：仅更改文档，例如更新README文件或者注释；
- perf：改进性能的代码更改，例如优化程序性能或者用户体验；
- refactor：代码更改，既不修复错误也不添加功能，例如重构代码以提高代码质量；
- style：不改变代码逻辑，仅修改代码风格，例如更改空格、格式化、分号等；
- test：添加缺失的测试或更正现有测试，例如编写单元测试、集成测试等；
- revert：回滚到某一版本，例如回滚到v1.1.8版本。

subject说明

以下是subject注意事项：

- 撰写描述时，尽量简短，一句话概括意思；

- 主要使用中文，确保信息易被广泛理解；
- 内容简短，无需增加符号。

例如

- feat: 迁移支持加密到普通；
- fix: 删除迁移过程的敏感日志；
- chore: 修改环境变量配置。

4. 账户名/邮箱规定

账户名：姓名的拼音，比如：huzhengxing，

邮箱：公司邮箱，比如：zhengxing.hu0740@nvxclouds.com。