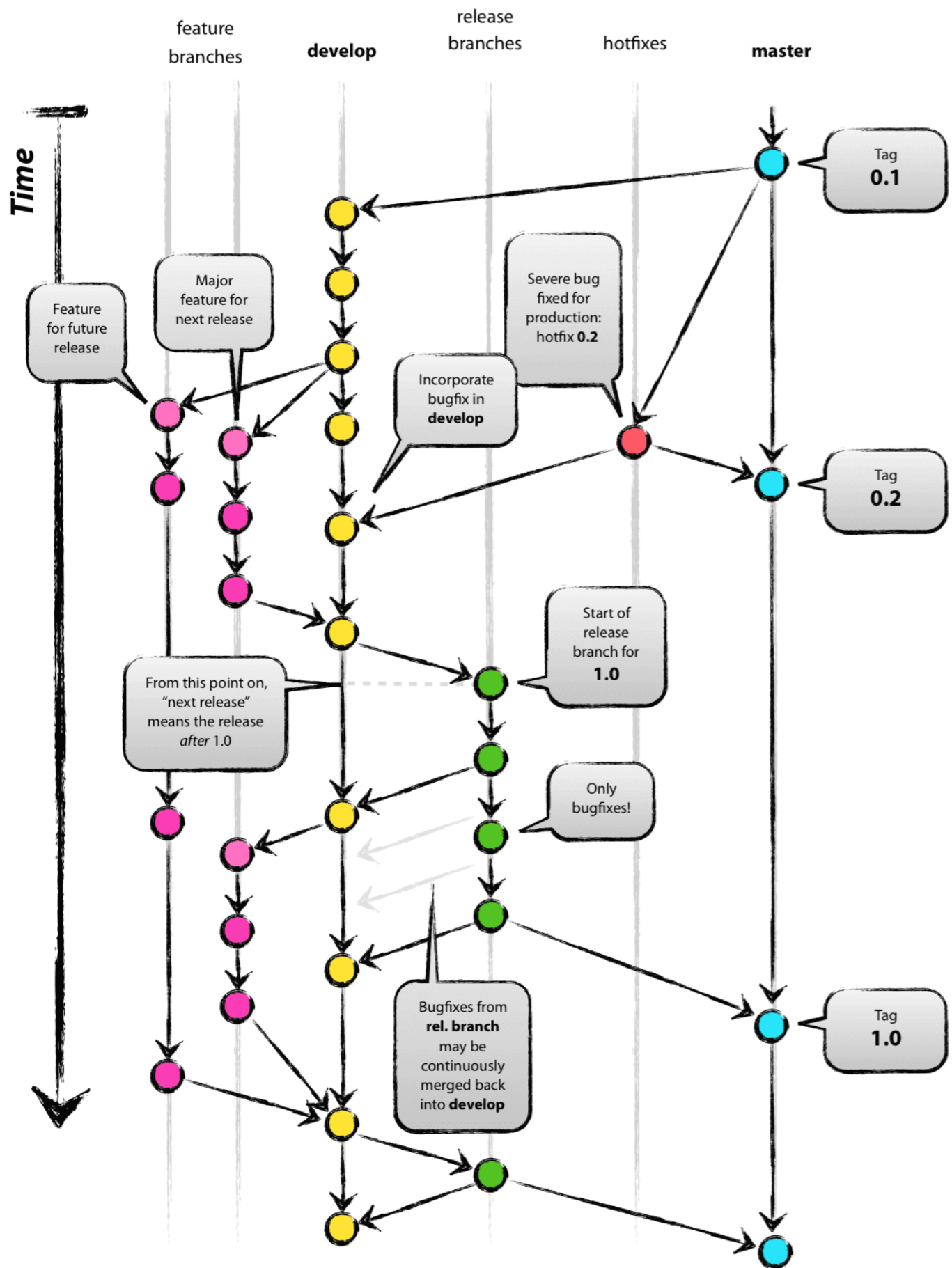


- Git Flow常用的分支
- Git Flow如何工作
 - 初始分支
 - 开发功能
 - 测试功能
 - 发布上线
 - 修复问题
 - 分支命名
 - 发布日期
- 常用git flow命令



Git Flow常用的分支

- **Master** 这个分支包含最近发布到生产环境的代码，最近发布的Release，这个分支只能从其他分支合并，不能在这个分支直接修改

只允许存在一个**Master**分支，并且需要保护起来，只要小组负责人才有合并权限。

- **Develop** 这个分支在项目开始时创建并在整个开发过程中维护，并包含具有正在测试过程中的新开发功能的预生产代码

新创建的功能应该基于开发分支，然后在准备好进行测试时合并回来。

只允许存在一个**Develop**分支，并且需要保护起来，只有小组负责人才有合并权限。

- **Feature** 这个分支主要是用来开发一个新的功能，功能开发完成并且通过**Code Review**后，合并回**Develop**分支，并进入下一个**Release**
- **Release** 当你需要发布一个新**Release**的时候，我们基于**Develop**分支创建一个**Release**分支，完成**Release**后，我们合并到**Master**和**Develop**分支。通常，在**Release**分支上执行的工作涉及与发布新代码相关的收尾工作和小错误。
- **Hotfix**

当发现线上环境的代码有小问题或者做些文案修改时，相关开发人员就在本地创建**hotfix** 分支进行修改。

如果是相当严重的问题，可能就得回滚到上一个 **tag** 的版本了。

Git Flow如何工作

初始分支



开发功能

在确定发布日期之后，将需要完成的内容细分一下分配出去，负责某个功能的开发人员利用 **SourceTree** 所提供的 **Git Flow** 工具创建一个对应的 **feature** 分支。

功能开发完并自测之后，先切换到 **develop** 分支将最新的代码拉取下来，再切换回自己负责的 **feature** 分支把 **develop** 分支的代码合并进来。如果有冲突则自己和配合的人一

起解决。

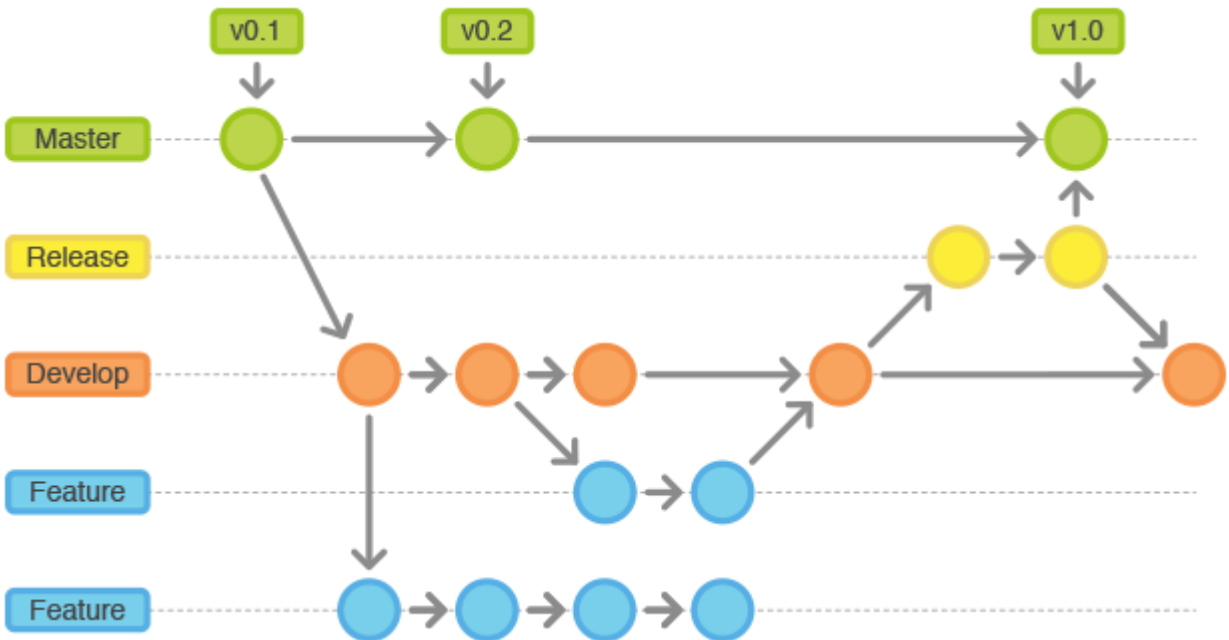
然后，到 **GitLab** 上的项目首页创建合并请求（**merge request**）。

项目负责人在收到合并请求时，应该先做下代码审核看看有没有明显的严重的错误；有问题就找负责开发的人去修改，没有就接受请求并删除对应的 **feature** 分支。



测试功能

负责测试的人创建一个 **release** 分支部署到测试环境进行测试；若发现了 **bug**，相应的开发人员就在 **release** 分支上或者基于 **release** 分支创建一个分支进行修复。



发布上线

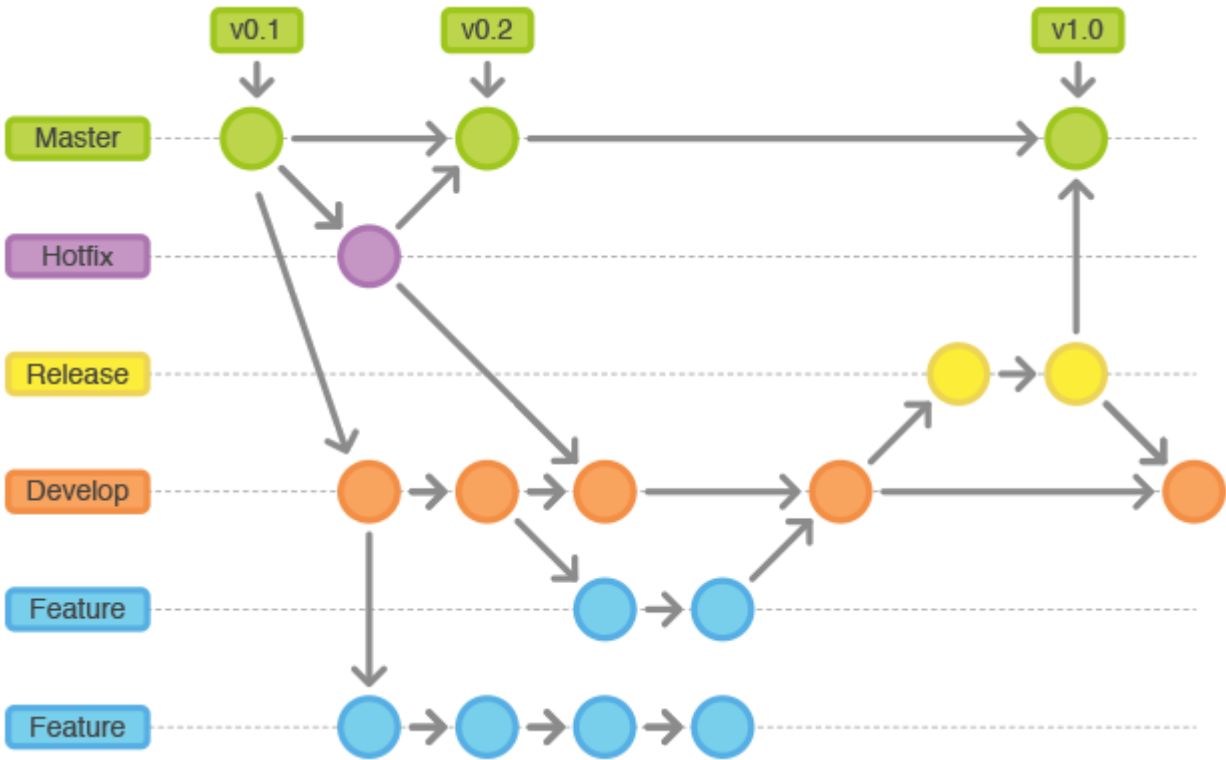
当确保某次发布的功能可以发布时，负责发布的人将 **release** 分支合并进 **master** 和 **develop** 并打上 **tag**，然后打包发布到线上环境。

建议打 **tag** 时在信息中详细描述这次发布的内容，如：添加了哪些功能，修复了什么问题。

修复问题

当发现线上环境的代码有小问题或者做些文案修改时，相关开发人员就在本地创建 **hotfix** 分支进行修改。

如果是相当严重的问题，可能就得回滚到上一个 **tag** 的版本了。



分支命名

除了主要分支的名字是固定的之外，派生分支是需要自己命名的，这里就要有个命名规范了。强烈推荐用如下形式：

- **feature**——按照功能点命名；
- **release**——用发布时间命名，可以加上适当的前缀；
- **hotfix**——Jira 的 **issue** 编号或 **bug** 性质等。

另外还有 **tag**，用语义化的版本号命名。

发布日期

发布频率是影响开发人员与测试人员的新陈代谢和心情的重要因素之一，频繁无规律的发布会导致内分泌失调、情绪暴躁，致使爆粗口、砸电脑等状况出现。所以，确保一个固定的发布周期至关重要！

在有一波或几波需求来临之时，想挡掉是不太可能的，但可以在评审时将它（们）分期，在某个发布日之前只做一部分。这是必须要控制住的！不然任由着需求方说「这个今天一定要上」「那个明天急着用」的话，技术人员就等着进医院吧！

工具推荐：

SourceTree (可以实际演示一下)

常用git flow命令

```
$ git flow feature start rss-feed
Switched to a new branch 'feature/rss-feed'
```

Summary of actions:

- A new branch 'feature/rss-feed' was created, based on 'develop'
- You are now on branch 'feature/rss-feed'

```
$ git flow feature finish rss-feed
Switched to branch 'develop'
Updating 6bcf266..41748ad
Fast-forward
   feed.xml | 0
   1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 feed.xml
Deleted branch feature/rss-feed (was 41748ad).
```

```
$ git flow release start 1.1.5
Switched to a new branch 'release/1.1.5'
```

```
$ git flow release finish 1.1.5
```

```
$ git flow hotfix start missing-link
```

```
$ git flow hotfix finish missing-link
```

