Br ne  
 *brain controlled drone*

System Configuration

**Purpose**

This Document’s purpose is to provide a brief overview of the used versioning techniques and elaborate on why they were chosen.

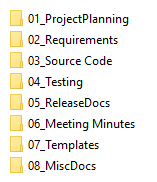
**Which tools do we use?**

We are going to use the school intern GitLab server. Due to our collaboration with the company g.tec, we are working with important data, such as the company’s libraries, which are not intended to be publicly accessible. The used GitLab - considering it providing private Repositories - is a fitting platform for the Brone Project files. Furthermore, a Premium GitHub account was taken into consideration due to it also providing private Repositories. After evaluating all the positive and negative aspects of each possibility, school-owned servers and direct communication to the administrators were the determining arguments for using GitLab. The client application for managing project files will be Source Tree, which provides all crucial features and is also accustomed to all group members, which is expected to result in a more trouble-free work.

**Which documents are under version control?**

All Brone project files, including Documents, Prototypes and the project itself will be under version control. We were not able to find any reason why we should exclude any files from version control.

**Organization of artifacts**

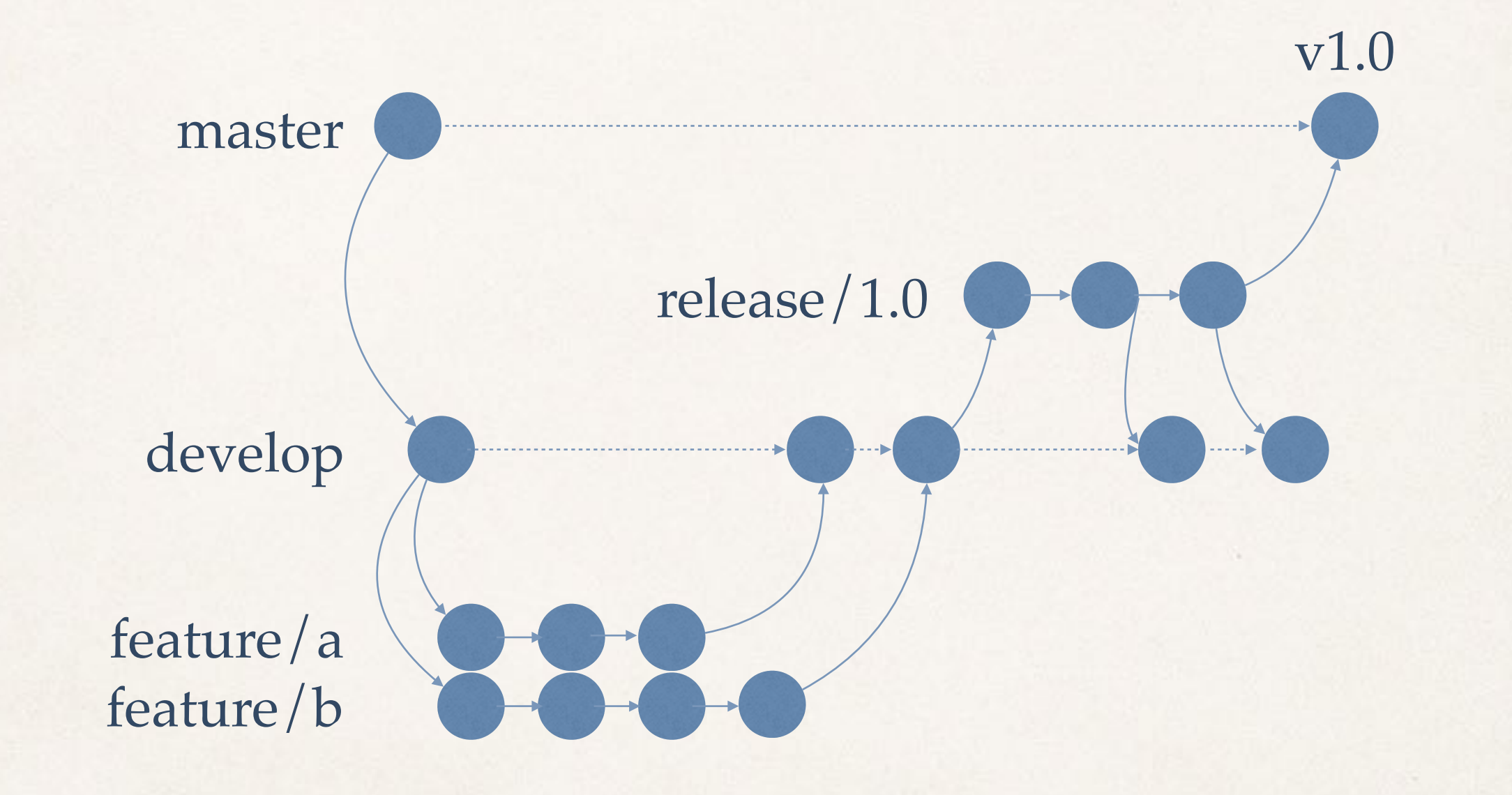
****

We are going to use this folder structure. The reason for this is simple. At first it is very structured so we know where the searched file is. Furthermore, we can divide files exactly into folders, because we have every important project area covered.

|  |  |
| --- | --- |
| Name | Function |
| ProjectPlannig | Files for project planning are stored in this folder. As example: Project Proposal, System Specs. |
| Requirements |  |
| Source Code | In this folder, we store our whole project and all prototypes. |
| Testing | Test-files for testing our project. |
| ReleaseDocs |  |
| Meeting Minutes | Take areas off every project worker. Also, information about meetings. |
| Templates | Templates for android or other programming stuff. |
| MiscDocs | Document we can´t divide into the above folder. |

**Variant management**

The following diagram displays our guidelines for branching. It is elaborated further down below.



At the beginning of our project, the “master”-Branch as well as a “develop”-Branch are created.

Let us roll up the concept from back to forth and therefore begin with the branches created when developing features. Each time a new feature starts being worked on, a new branch for it is created. When it is ready for the project-group to work with it is pushed to the develop branch. When the project is once again ready for a release, a branch where the version gets checked for various bugs is created. When done doing so, it is pushed to the develop branch (for further development) as well as the master branch.