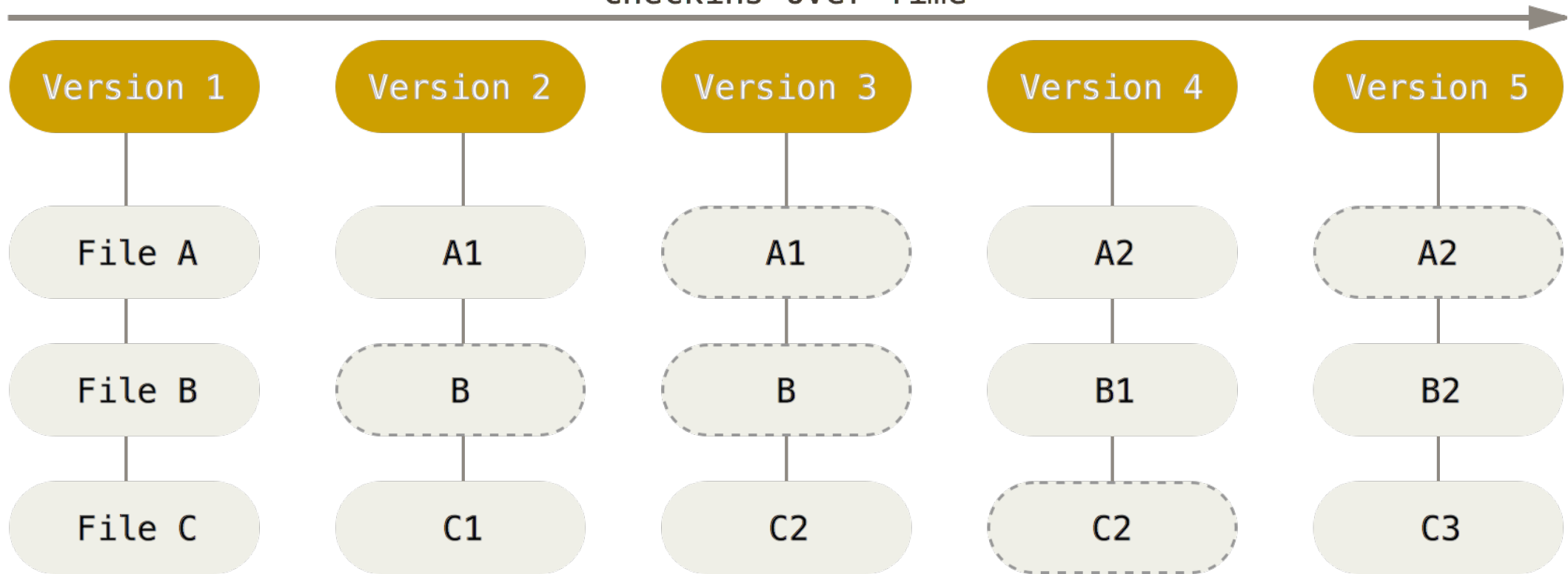


GIT

- Desenvolvimento iniciado depois de um rage do Linus com a empresa BitKeeper
- Foco no suporte a diversas branches paralelas

GIT

Checkins Over Time



GIT

Working
Directory

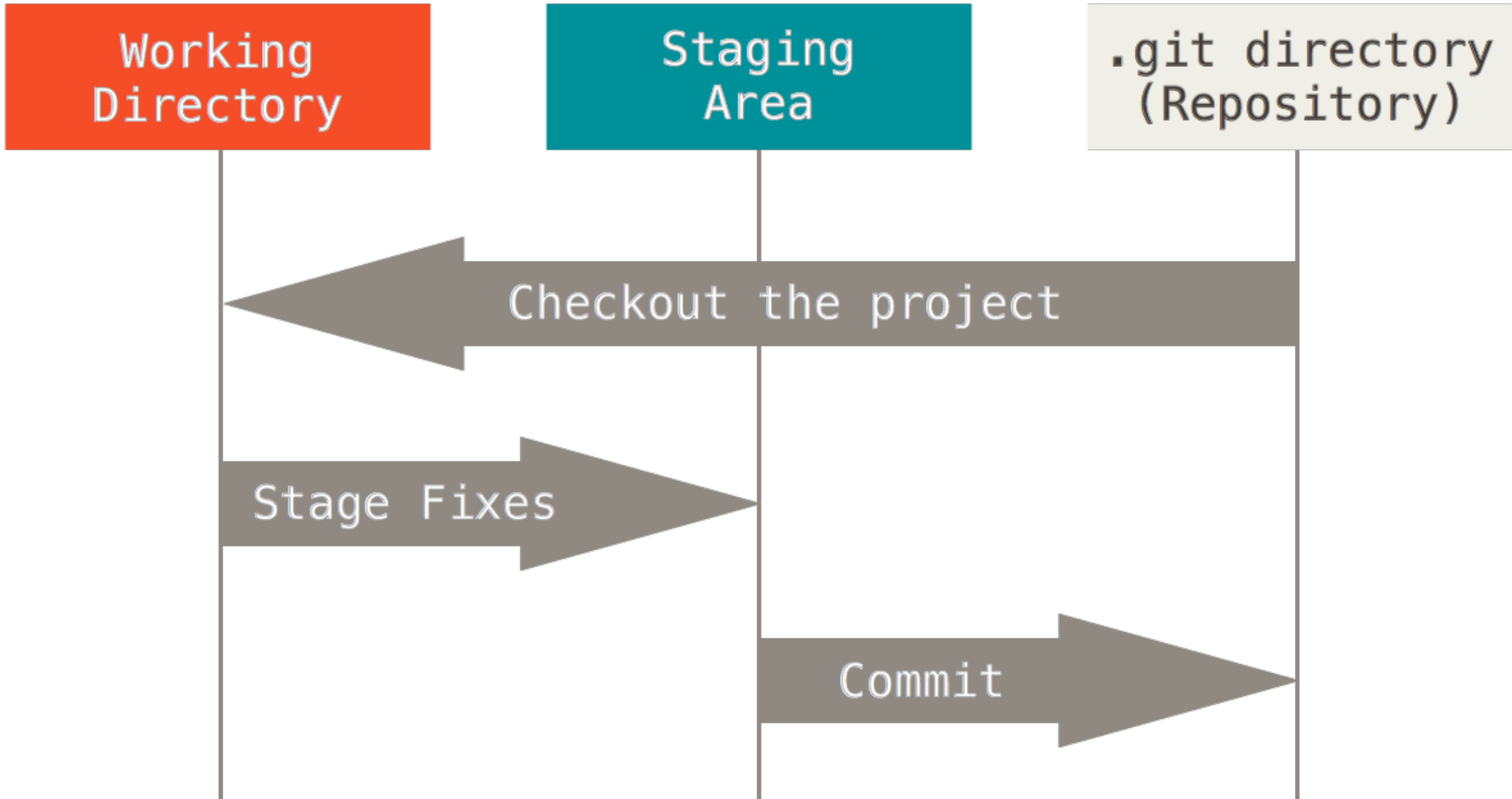
Staging
Area

.git directory
(Repository)

Checkout the project

Stage Fixes

Commit



GIT

- 1) Modifique (criar, deletar, editar)
- 2) Adicione os arquivos modificados na área de preparação (staging/index)
- 3) Comite as mudanças na área de preparação (snapshot)

GIT



GIT

```
mkdir projeto
```

```
cd projeto
```

```
git init .
```

```
git status
```

```
git add <arquivo.txt>
```

```
git status
```

```
git commit -m "primeiro arquivo"
```

```
git status
```

GIT

- Repositórios GIT são locais!
- Exercício:
 - Use o comando `git clone` para clonar o seu repositório em outro diretório
 - `git help clone`

GIT

- Esqueci alguma coisa durante o commit!
 - Crie um arquivo “arquivo faltante.txt”
 - `git add “arquivo faltante.txt”`
 - `git commit --amend -m “nova msg para o log”`
 - `git log --oneline`

GIT

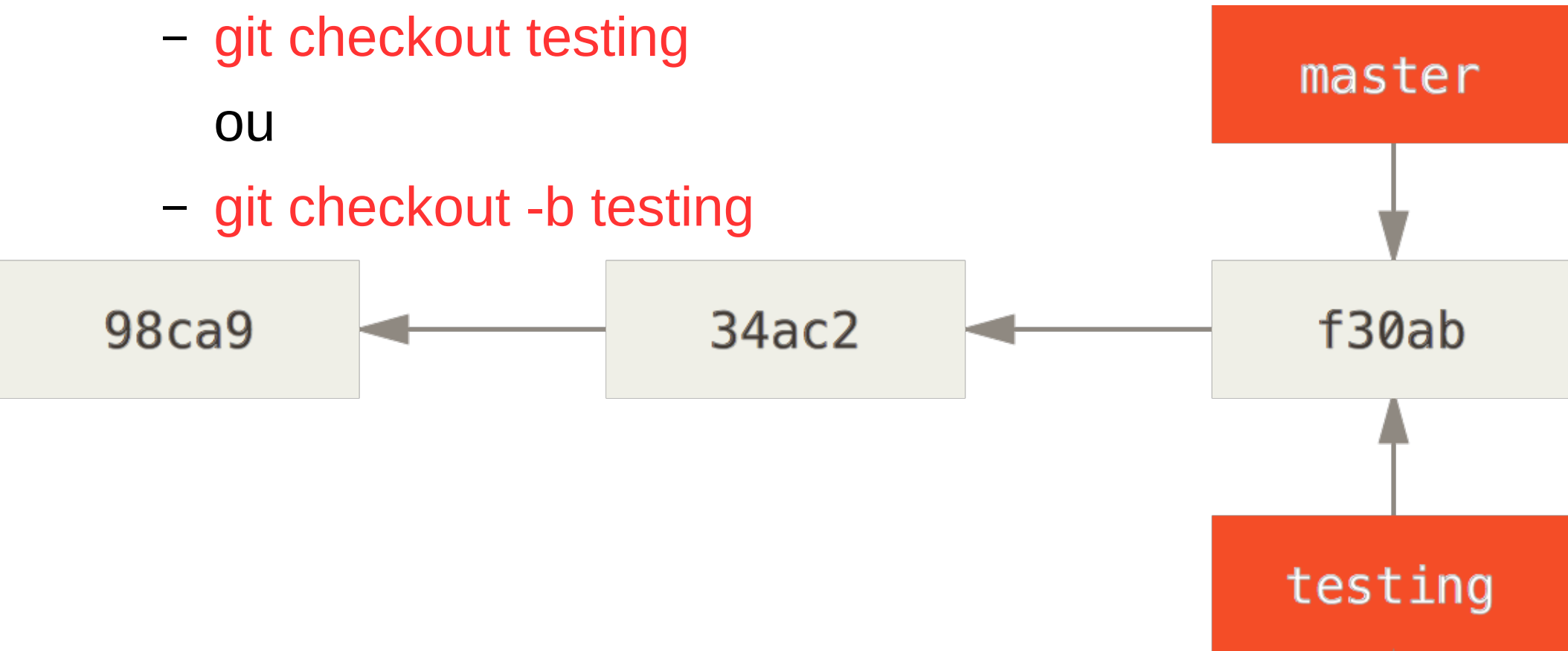
- Tags!
 - Tag no commit atual
 - `git tag -a v0.0.1 -m "first release unstable API"`
 - `git show`
 - Tag em um commit antigo
 - Encontre o hash do commit desejado
 - `git log --oneline`
 - TAG!
 - `git tag -a v0.0.1 -m "first release unstable API" 8a959`
 - Compartilhando tags
 - `git push origin --tags`

GIT

- Remotes!
 - Listando
 - `git remote -v`
 - Adicionando
 - `git remote add sibbr https://github.com/sibbr/projeto`
 - Removendo
 - `git remote remove sibbr`
 - Fetch: trazendo as mudanças, sem merge!
 - `git fetch sibbr`
 - Pull: trazendo as mudanças, com merge!
 - `git pull`

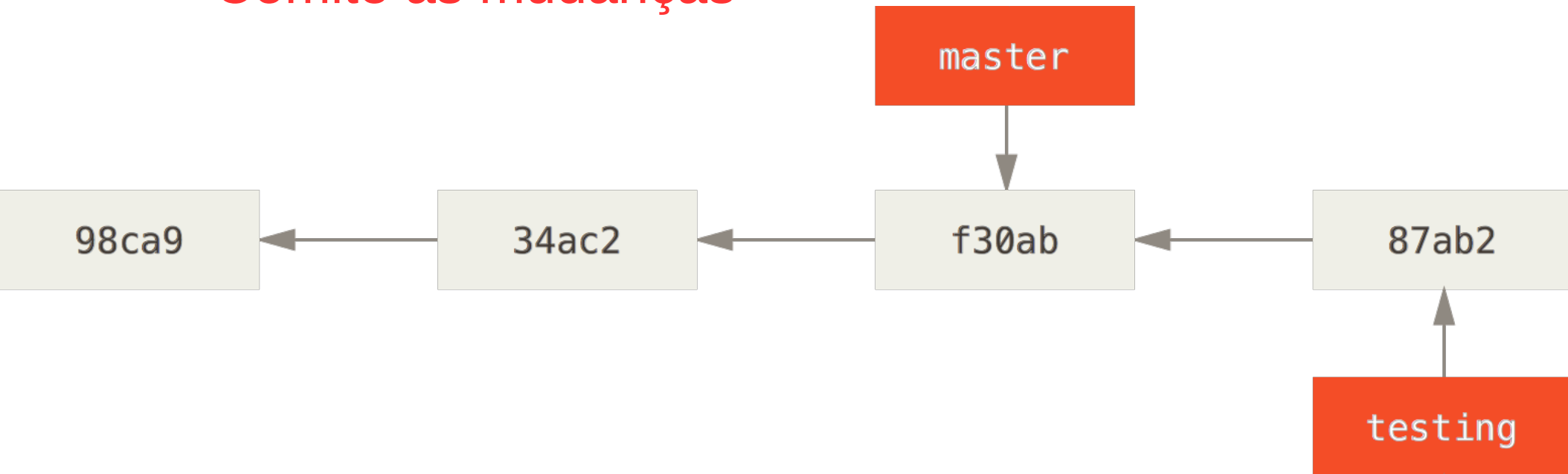
GIT

- Branches!
 - git branch testing
 - git checkout testing
 - ou
 - git checkout -b testing



GIT

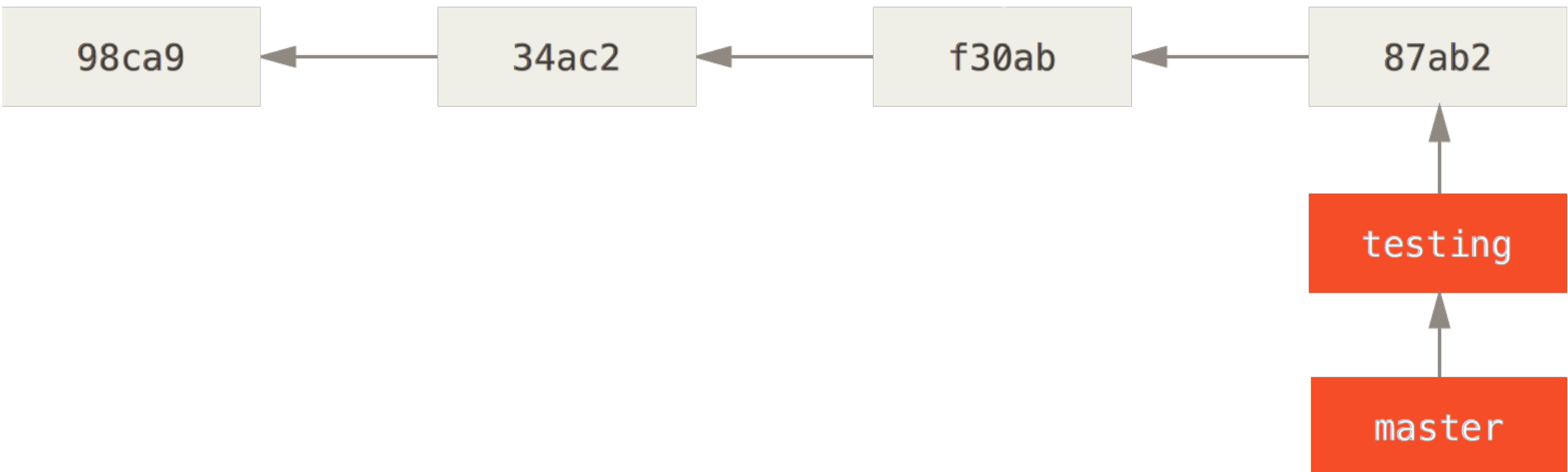
- Branches!
 - Modifique algum arquivo
 - Comite as mudanças



GIT

- Branches!
 - `git merge testing`

Ancestral comum

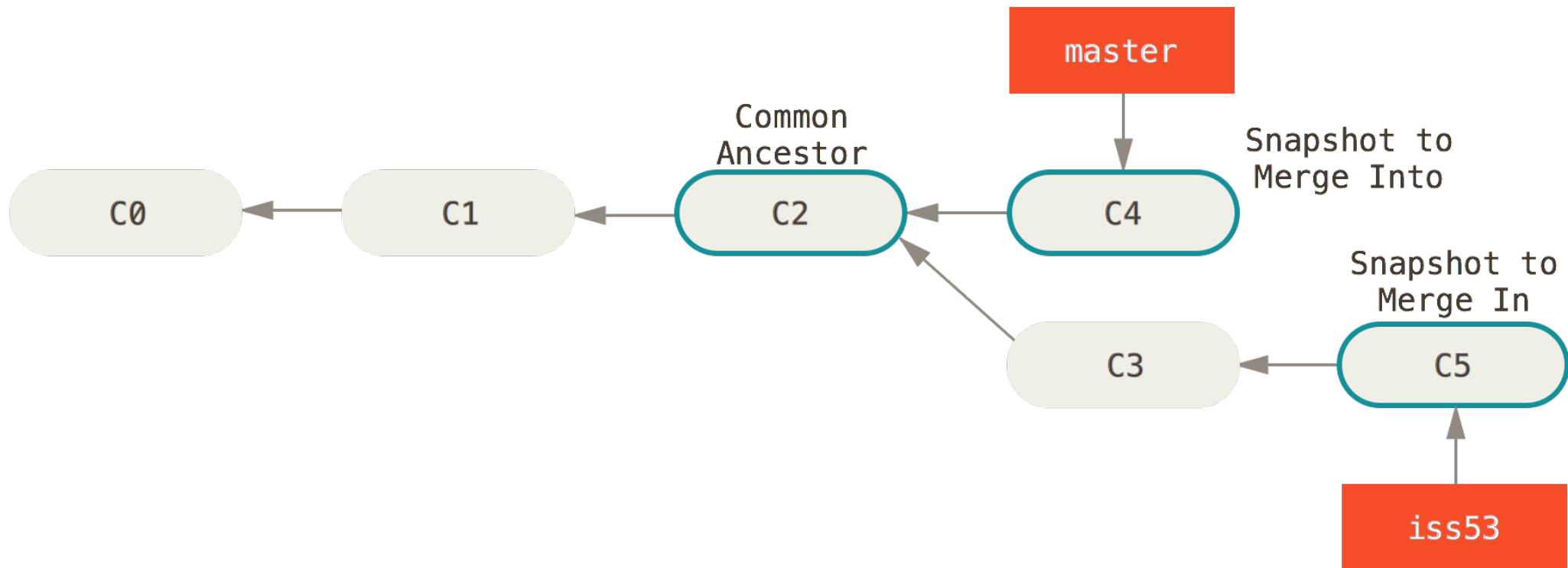


GIT

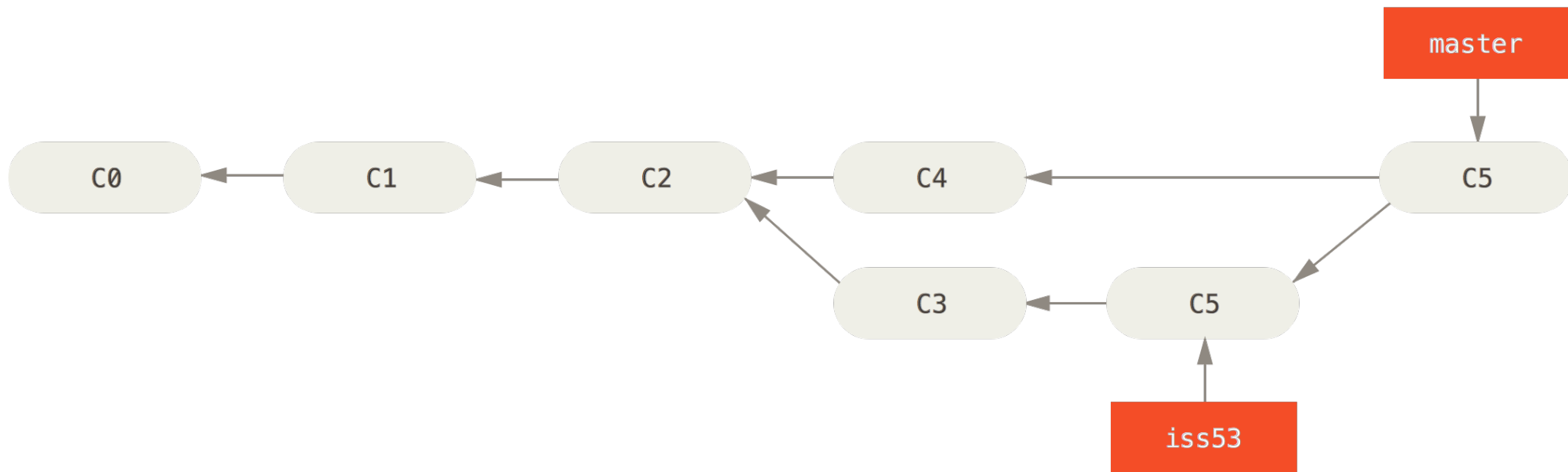
- Branches!

- 1) `git branch -d testing`
- 2) `git checkout -b iss53`
- 3) Crie um novo arquivo e comite
- 4) Faça o passo anterior novamente
- 5) Faça o passo 3 no master
- 6) `git merge iss53`

GIT



GIT



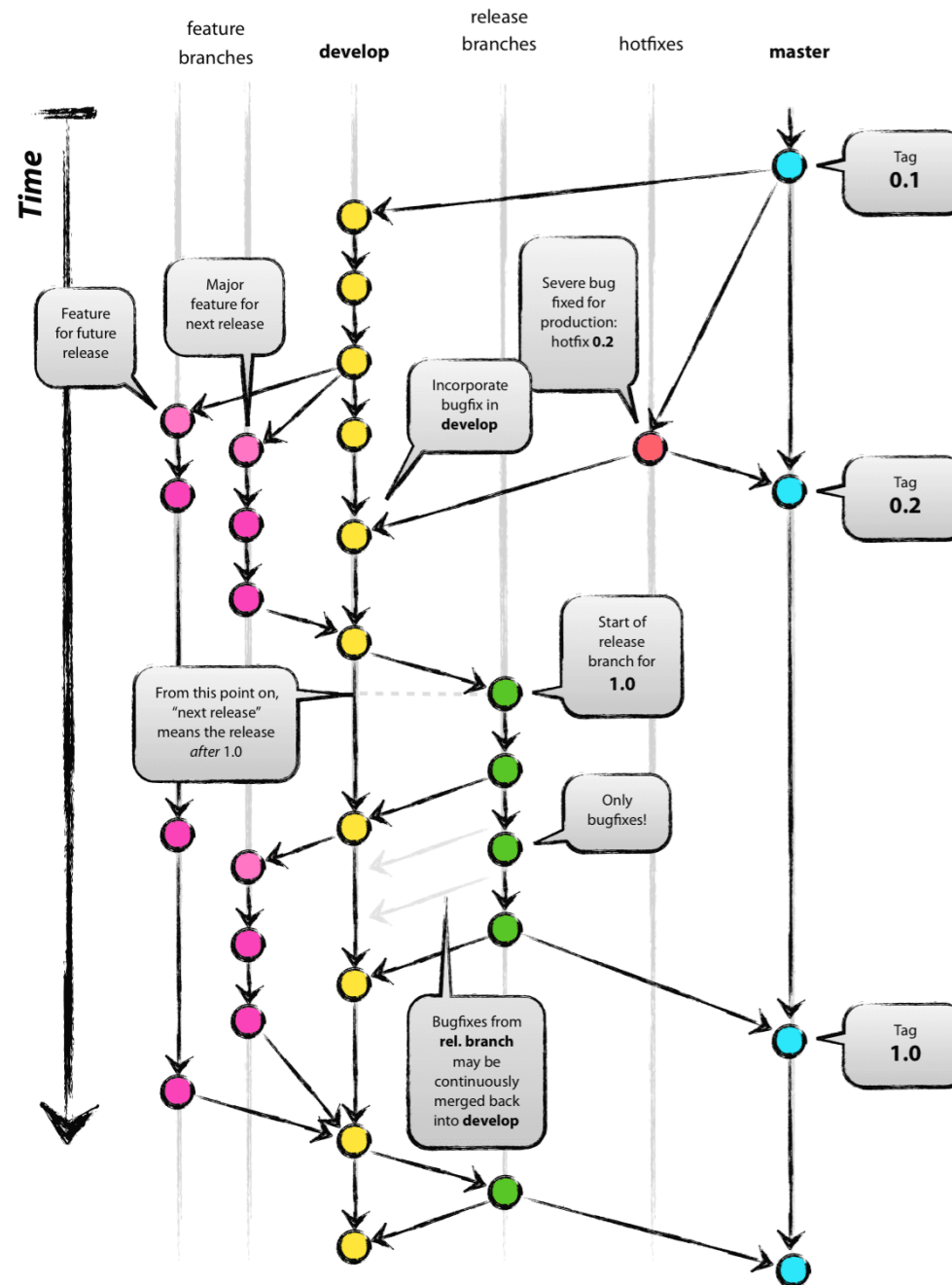
GIT

- Branches!
- Exercício de conflito:
 - Edite a mesma linha de um arquivo comum aos branches disponíveis
 - Tente realizar o merge e resolver os conflitos

GIT

- Branches!
 - Branching Workflow
 - Geralmente é definido para o projeto atual um modelo de branching que os desenvolvedores devem seguir;
 - Famoso (hackernews) post descrevendo um modelo e sendo criticado;
 - Leitura obrigatória: <http://nvie.com/posts/a-successful-git-branching-model/>

GIT



GIT

- Referências:
 - <https://git-scm.com/book/en/v2>
 - <http://nvie.com/posts/a-successful-git-branching-model/>
 - `$ git help <comando>`