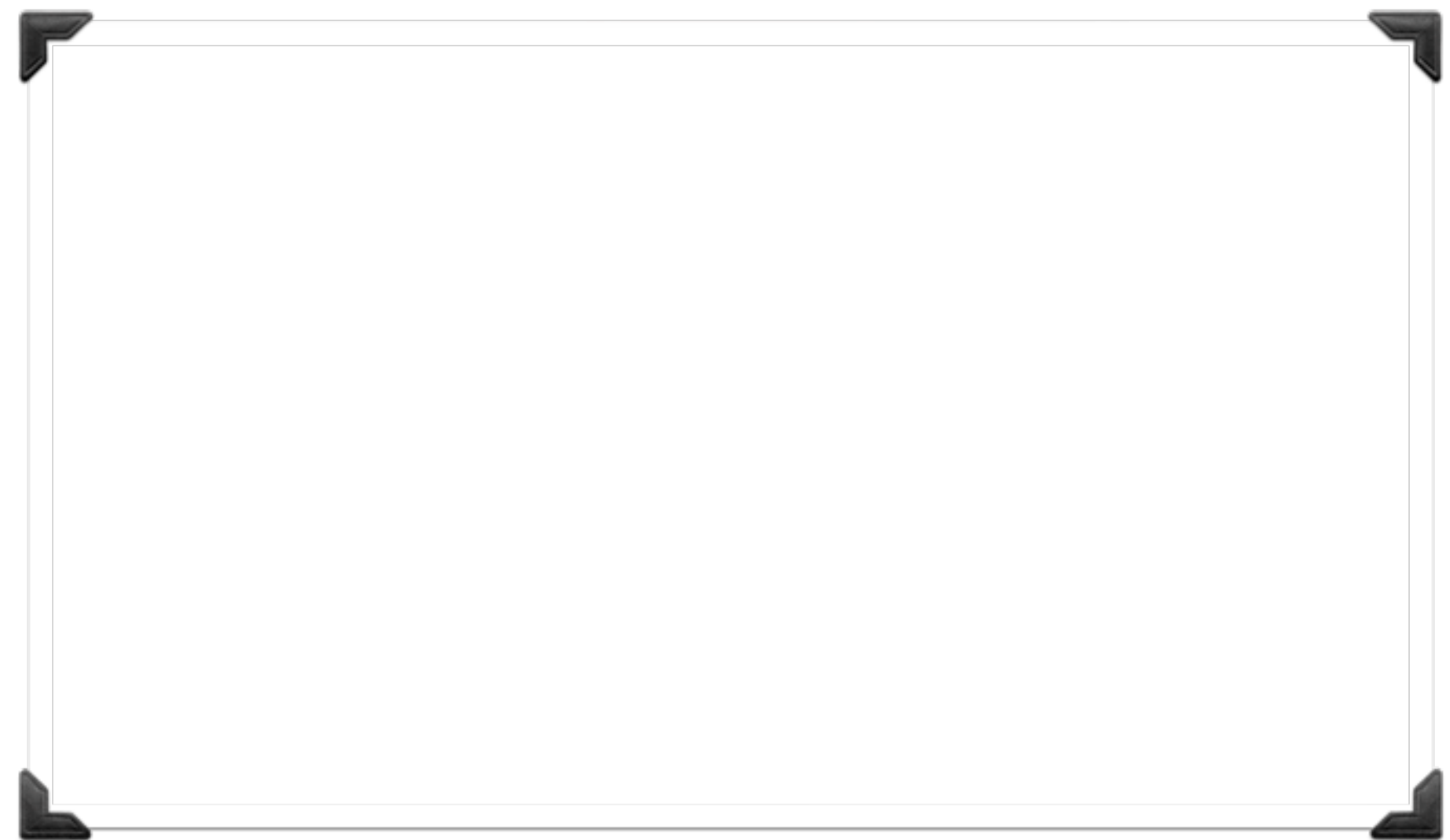


# Tools: Git

Designing and Maintaining Software (DAMS)

Louis Rose



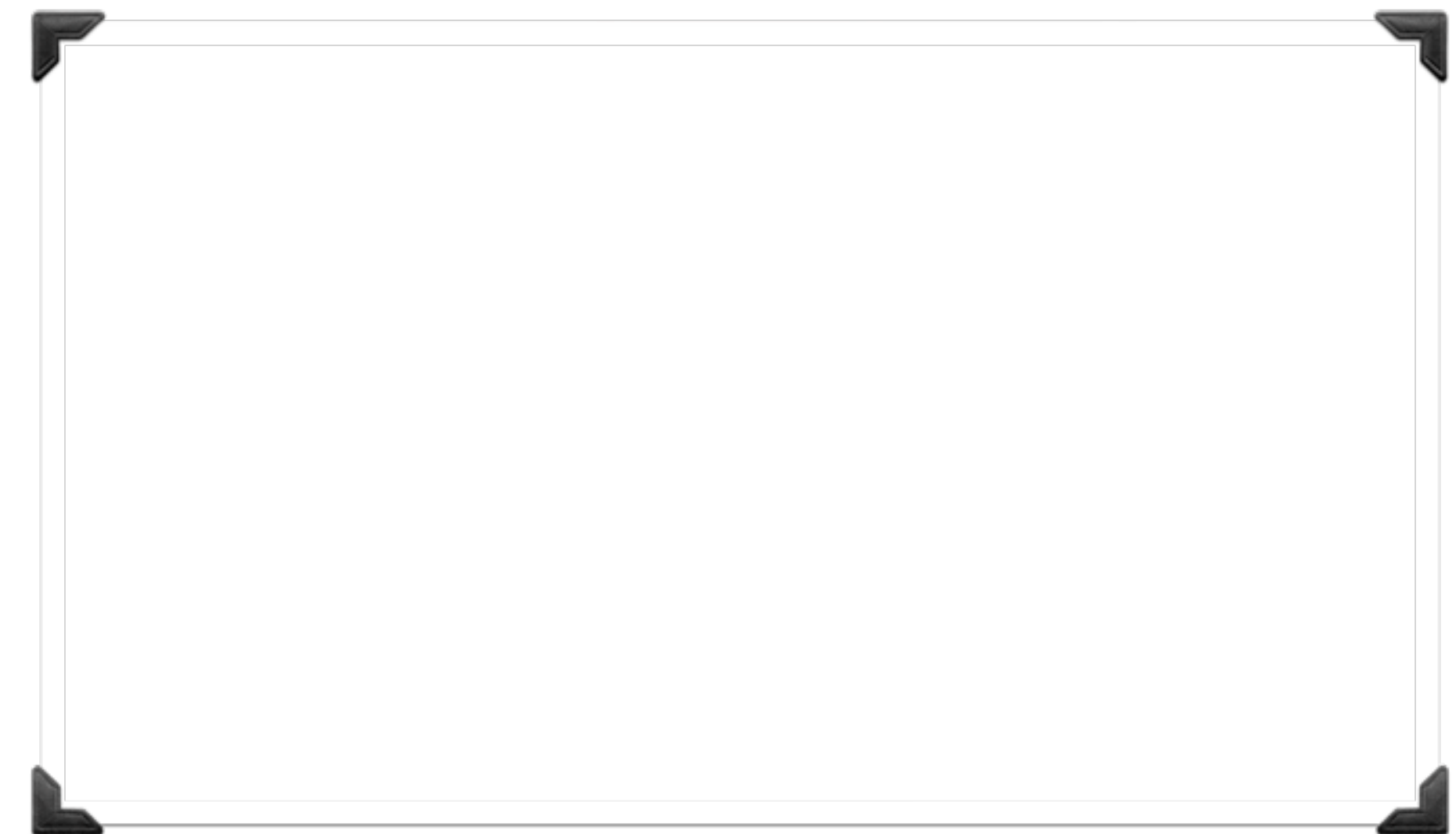
# Version control

To more easily track changes, collaborate, and find bugs, we use version control to manage software.

Return to Zero

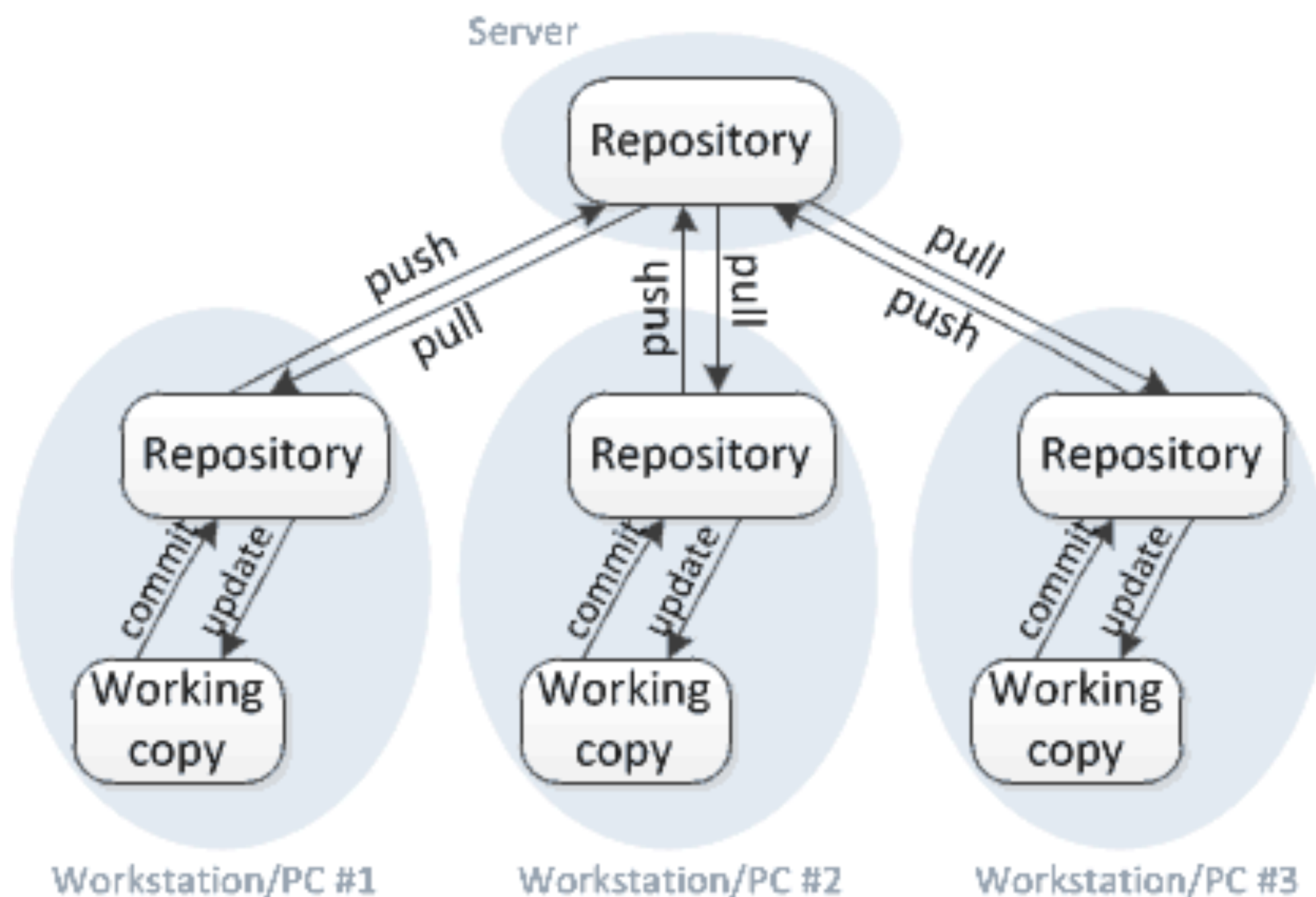


EEWeb.com

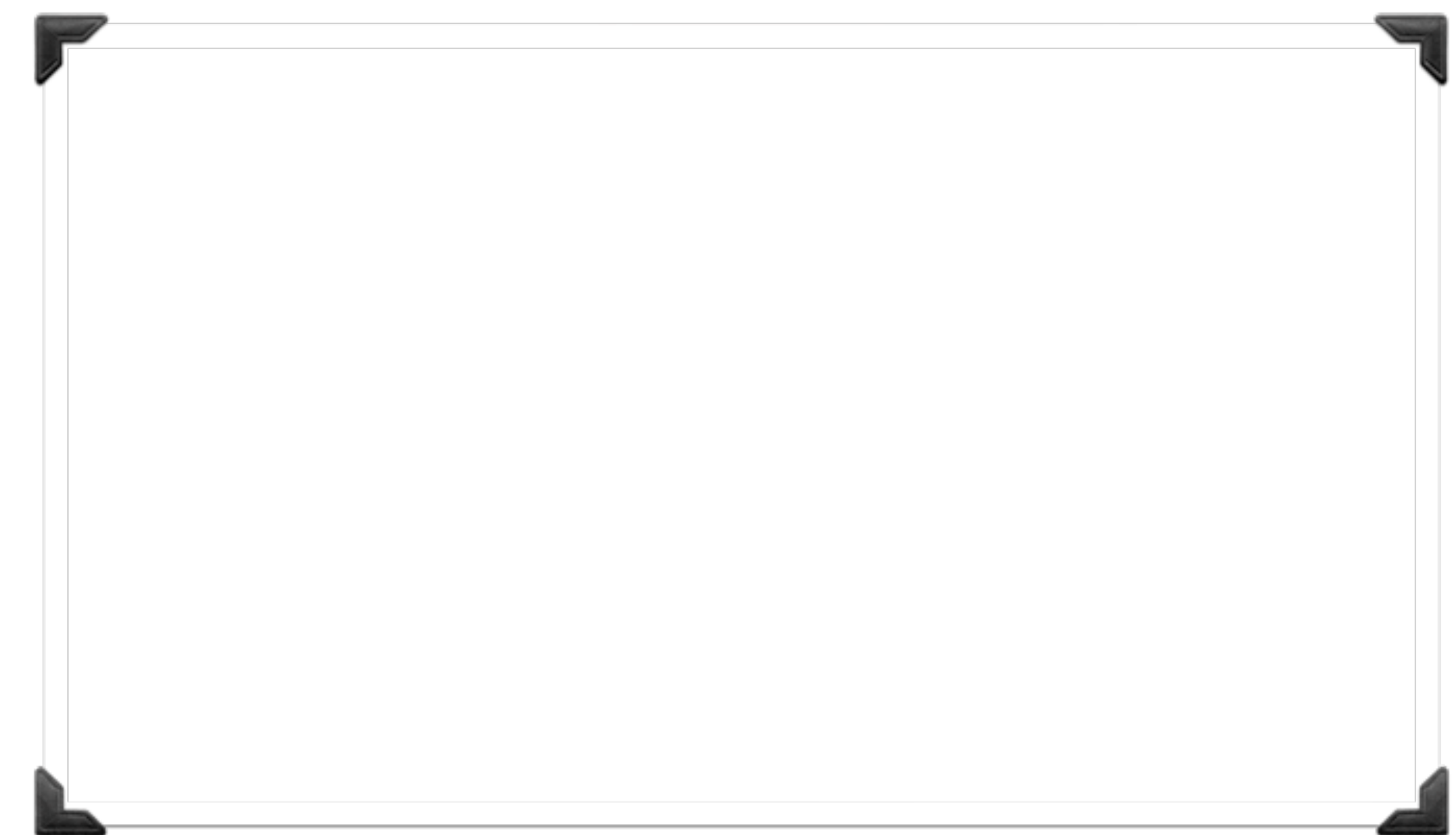


# Distributed version control

In which there need not be a canonical version of a repository: people can work independently, offline and branches are cheap.



<https://homes.cs.washington.edu/~mernst/advice/version-control.html>



# How much Git is needed for DAMS?

Staging changes and committing with **add** and **commit**

Sharing changes with **push**

Browsing history with **log** and **blame**

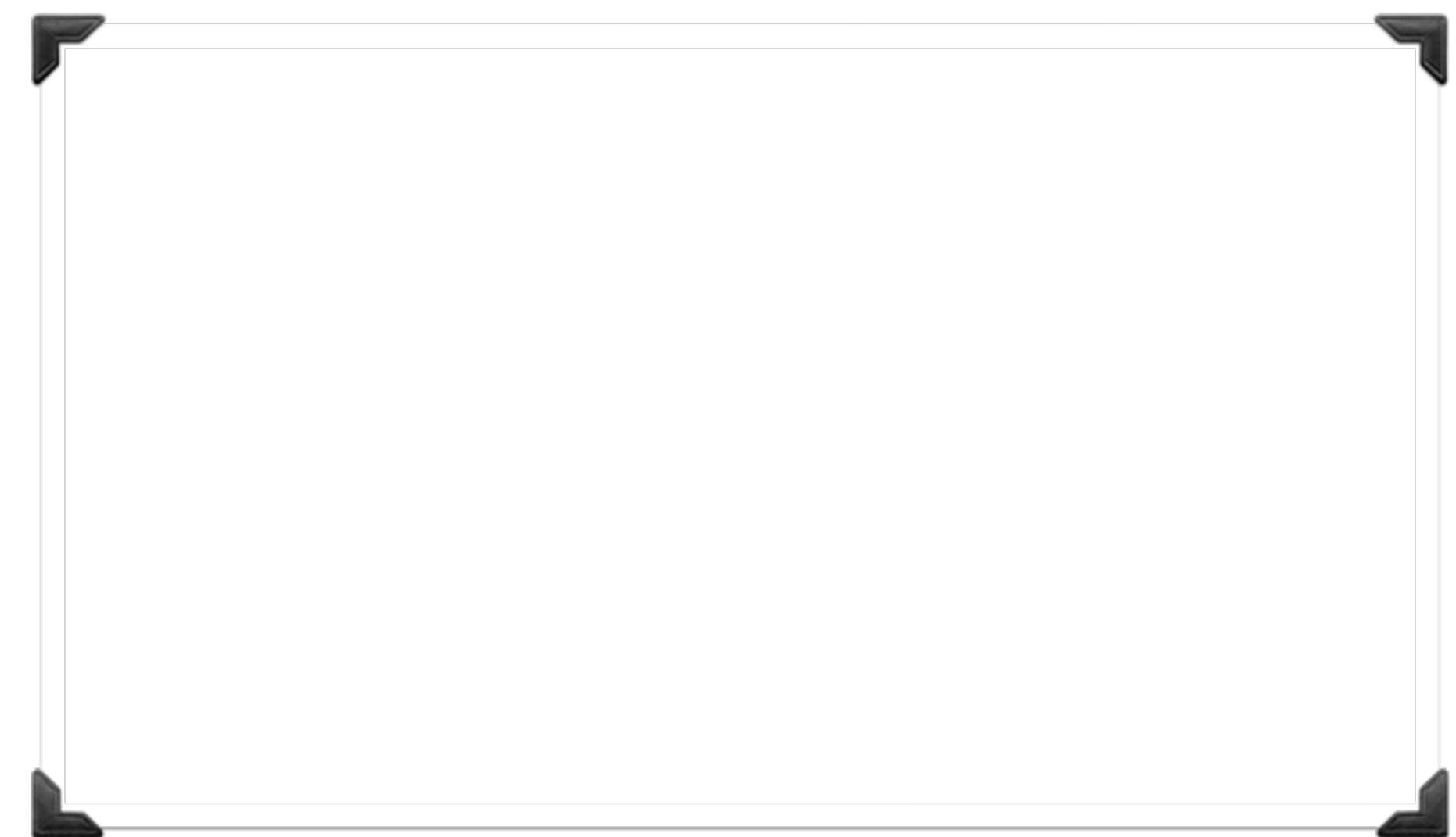
Combining changes with **fetch**, **merge** and **rebase**

Rewriting unshared history with interactive **rebase**



<http://www.git-scm.com/book>

<http://gitready.com>

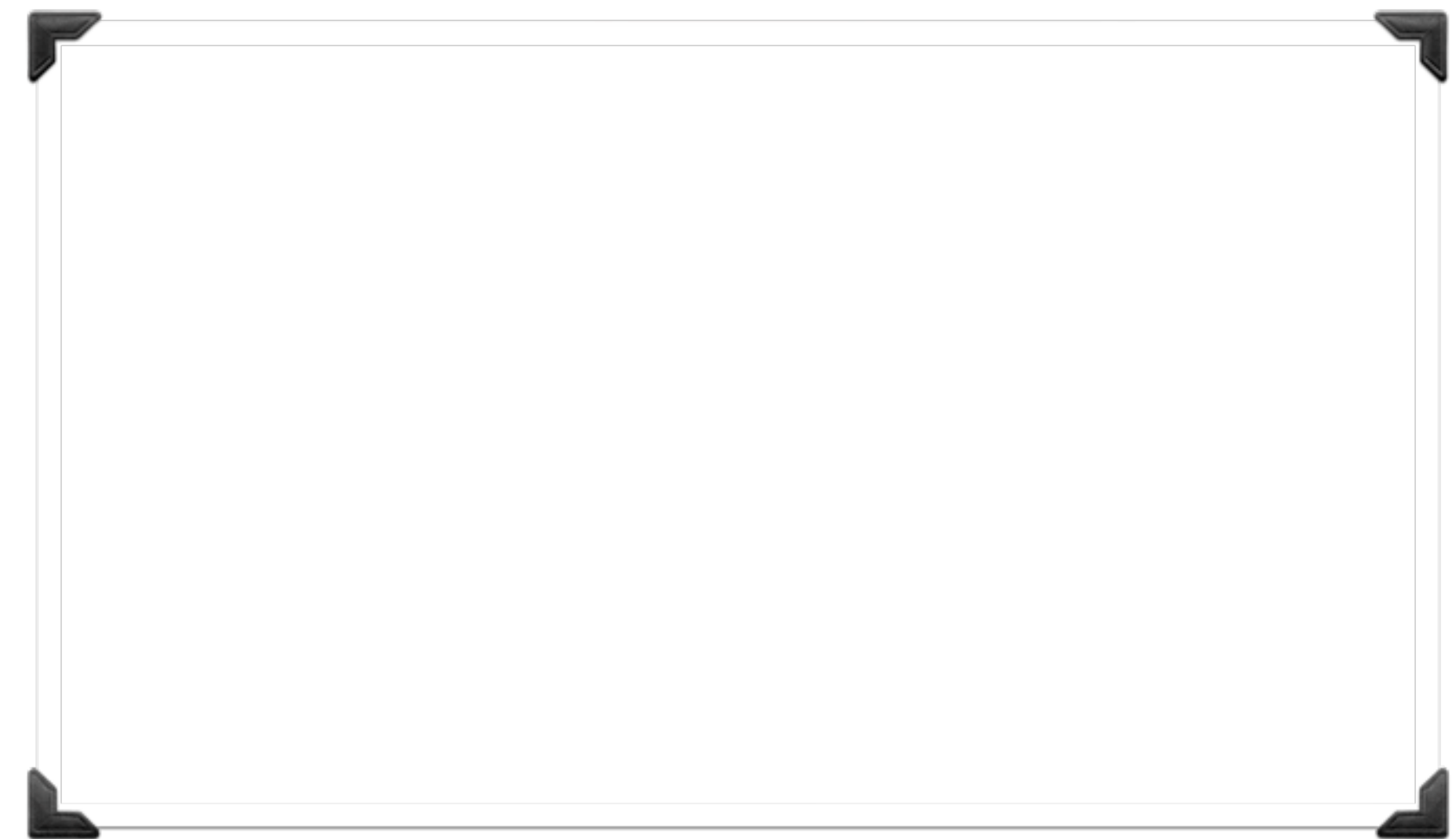


# Tip #1: Good commit messages

Commit messages provide useful context for your teammates (and for your future self).

So avoid this:

```
% git commit -m "Fix login bug"
```





# Tip #1: Good commit messages

Commit messages provide useful context for your teammates (and for your future self).

So avoid this:

```
% git commit -m "Fix login bug"
```

And prefer this:

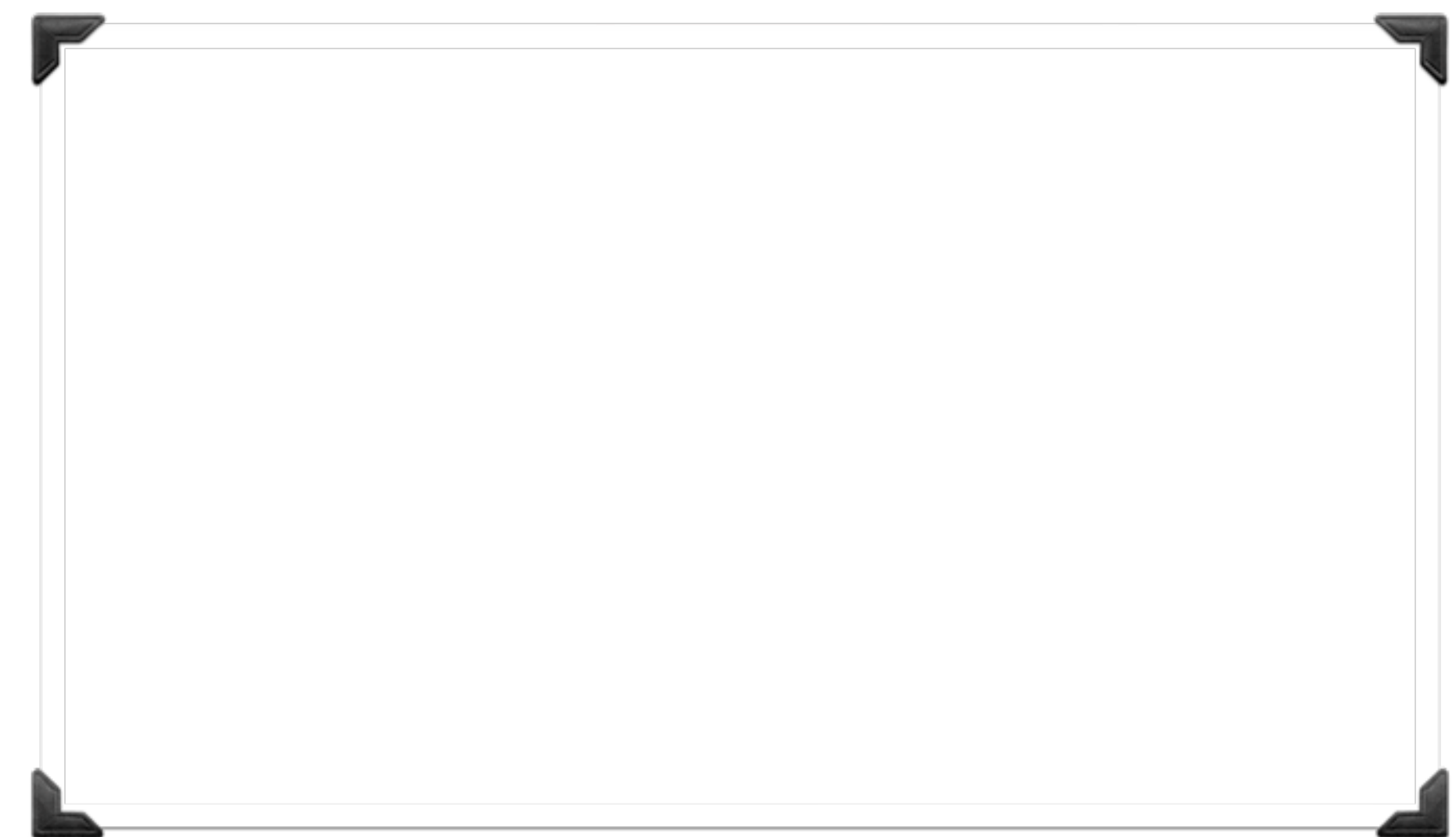
```
% git commit
```

```
Redirect user to the requested page after login
```

```
Fixes issue #42
```

```
Users were being redirected to the home page after login, which  
is less useful than redirecting to the page they had originally  
requested before being redirected to the login form.
```

```
* Store requested path in a session variable  
* Redirect to the stored location after successfully logging in  
the user
```



# Tip #1: Good commit messages

Commit messages provide useful context for your teammates (and for your future self).

A template:

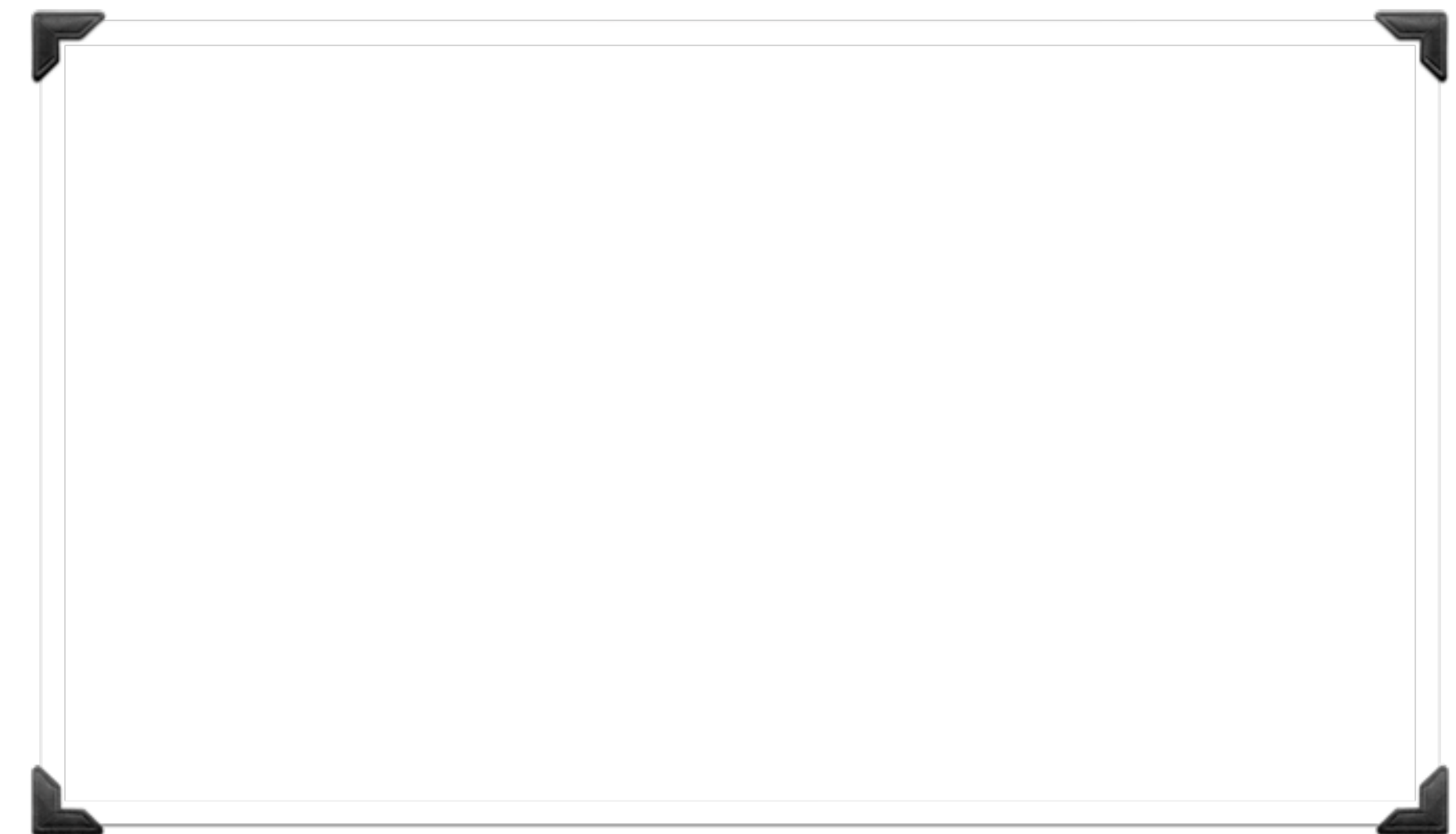
A short summary of why the change was necessary

Possibly a reference to an issue tracker

A high-level description of how the problem has been resolved, such as "Introduce a red/black tree to increase search speed."

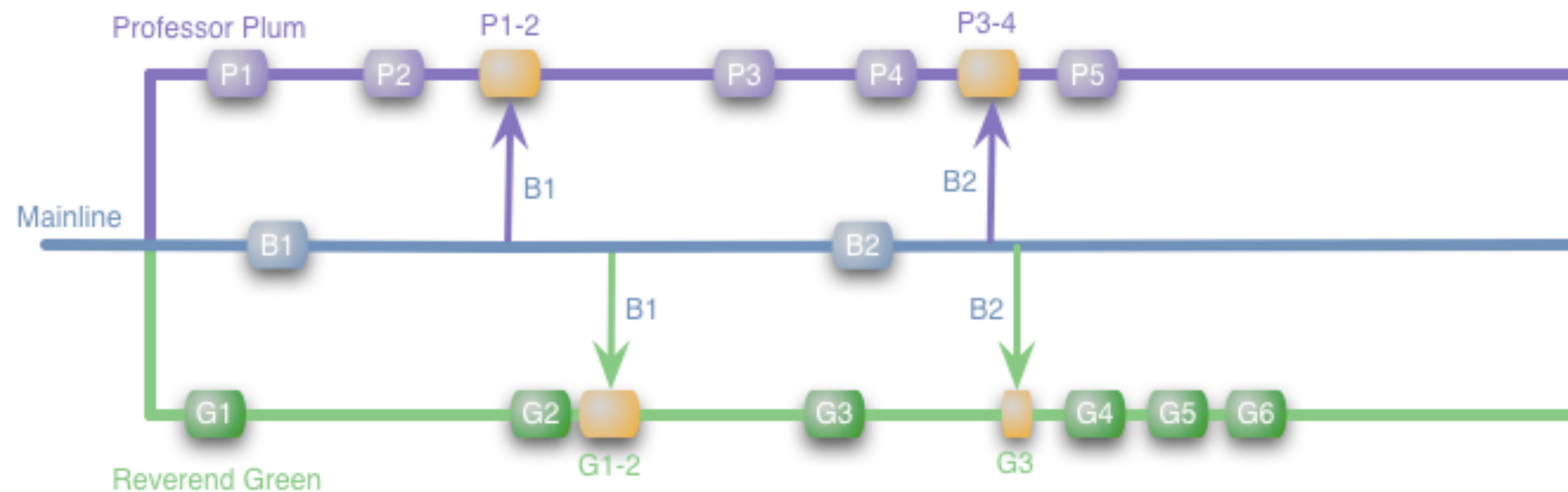
- \* First side-effect of this change
- \* Second side-effect of this change

<https://robots.thoughtbot.com/5-useful-tips-for-a-better-commit-message>



# Tip #2: Consider feature branches

Each feature is developed on a separate branch. Mainline changes are periodically merged into feature branches.



There are different strategies for merging feature branches into the mainline: e.g., only when the feature is finished, continuous integration or promiscuous integration.

