

# Git Ritual

Reuben Brasher

May 25, 2023

Why?

First steps

Next steps

Day to day

# Why?

Why git? Why PRs? Why the ritual?

```
git clone <my-fancy-project>
```

# Why?

Why git? Seriously?

```
git clone <my-fancy-project>
```

Because we said so.

# Why?

Why git? Seriously?

```
git clone <my-fancy-project>
```

~~Because we said so.~~

Because you said you would.

Because millions of people do.

Because it works.

Because it is powerful.

# Courage

You can be fearless!

- ▶ You can change

# Courage

You can be fearless!

- ▶ You can change
- ▶ You can change back

# Courage

You can be fearless!

- ▶ You can change
- ▶ You can change back
- ▶ You can delete



# Collaborative vs. Individual

You are not the only person in the world.

Your friends love you, but they do not have time to read all of your code and docs.

- ▶ PR and review

# Collaborative vs. Individual

You are not the only person in the world.

Your friends love you, but they do not have time to read all of your code and docs.

- ▶ PR and review
- ▶ History and blame

# Collaborative vs. Individual

You are not the only person in the world.

Your friends love you, but they do not have time to read all of your code and docs.

- ▶ PR and review
- ▶ History and blame
- ▶ Overall metrics

# Collaborative vs. Individual

You are not the only person in the world.

Your friends love you, but they do not have time to read all of your code and docs.

- ▶ PR and review
- ▶ History and blame
- ▶ Overall metrics
- ▶ Raise awareness of your work

# Establish an account

Use a registered email from your organization.

# Local identity

```
git config --global user.name "Your Name"  
git config --global user.email "your@email.com"
```

# SSH keys

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Please use a passphrase!

```
cat .ssh/id_ed25519.pub
```

# Clone existing repo

```
git clone <my-fancy-project>
```



# Create from scratch

Always have .gitignore

Always have README.md

Strongly consider LICENSE

Note that you are not just publishing current state, but commit history

Initially set repo as private

# Branch protection

Branch protection!

# Basic work

Always work in a branch

```
git checkout -b rb/my_cool_feature
```

Commit often

Commit changes to files in a descriptive manner

Commit may reference Jira

# PR

`git push`

Good commit description

Reference a jira ticket (No ticket?)

Request a review

# Merge PR

Resolve conflicts

Rebase or merge locally

Make sure tests continue to pass

# Feature branches?

Be cautious.

# References I