DevOps 2.0

The Path to Continuous Merge & Happy Developers

Yishai Beeri
CTO, LinearB

yishai@linearb.io

/:\ gitStream

Agenda

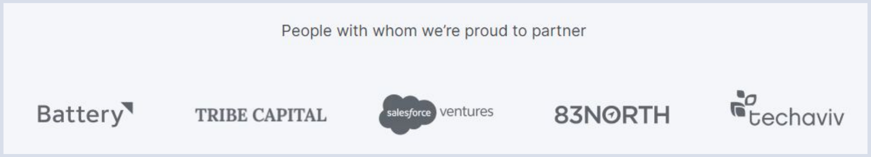
Background

What is Continuous Merge?

... and how can I use CM
to get my PRs merged faster?

LinearB

The Software Delivery Management Platform for Engineering Leaders



People with whom we're proud to partner

Battery TRIBE CAPITAL salesforce ventures 83NORTH techaviv

DRATA BigID Appcues TOMTOM EQUINIX

hotjar ClickUp RED VENTURES TaskRabbit MURAL

A Brief History of Software Development

Waterfall

Giant PRD and you build what your manager wants
Release Cycle - 9 - 24 months

Agile

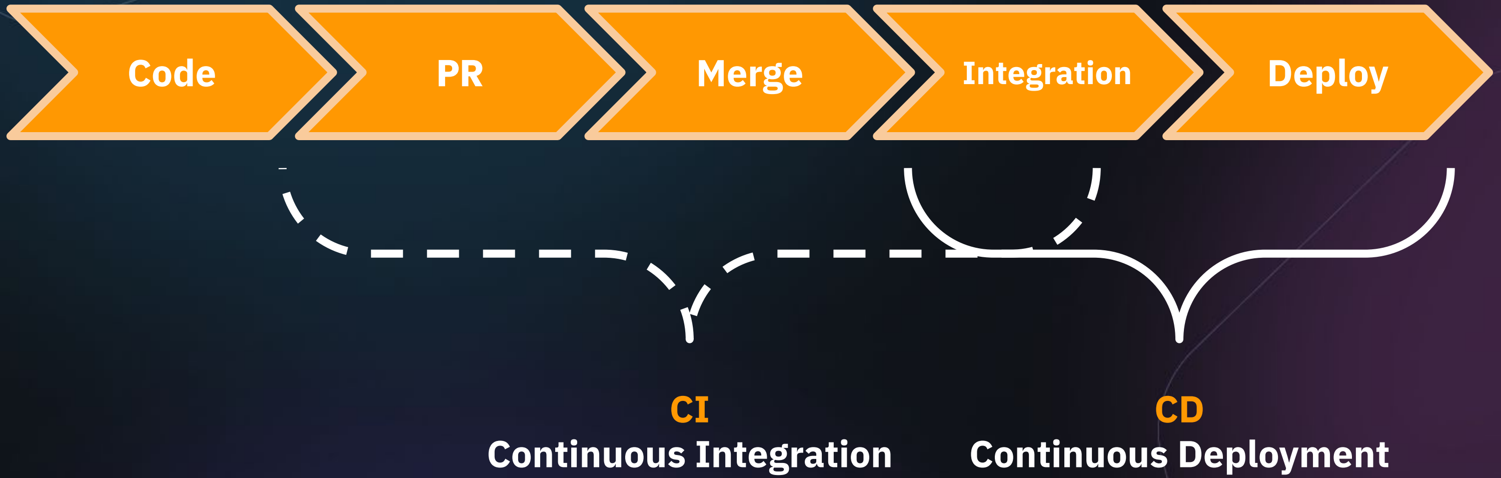
Sprints! Story Points!
Standups!
5000 arguments on what is Agile and what is not.
Toss it over: its an ops problem, works for me.

DevOps

I need to run what I write?
Pull Request for code review to see if its ready for testing.

Unit tests, user tests, tests for tests and more tests. Tests pass? Great! Deploy or Deliver.

CI / CD approach



Code

PR

Merge

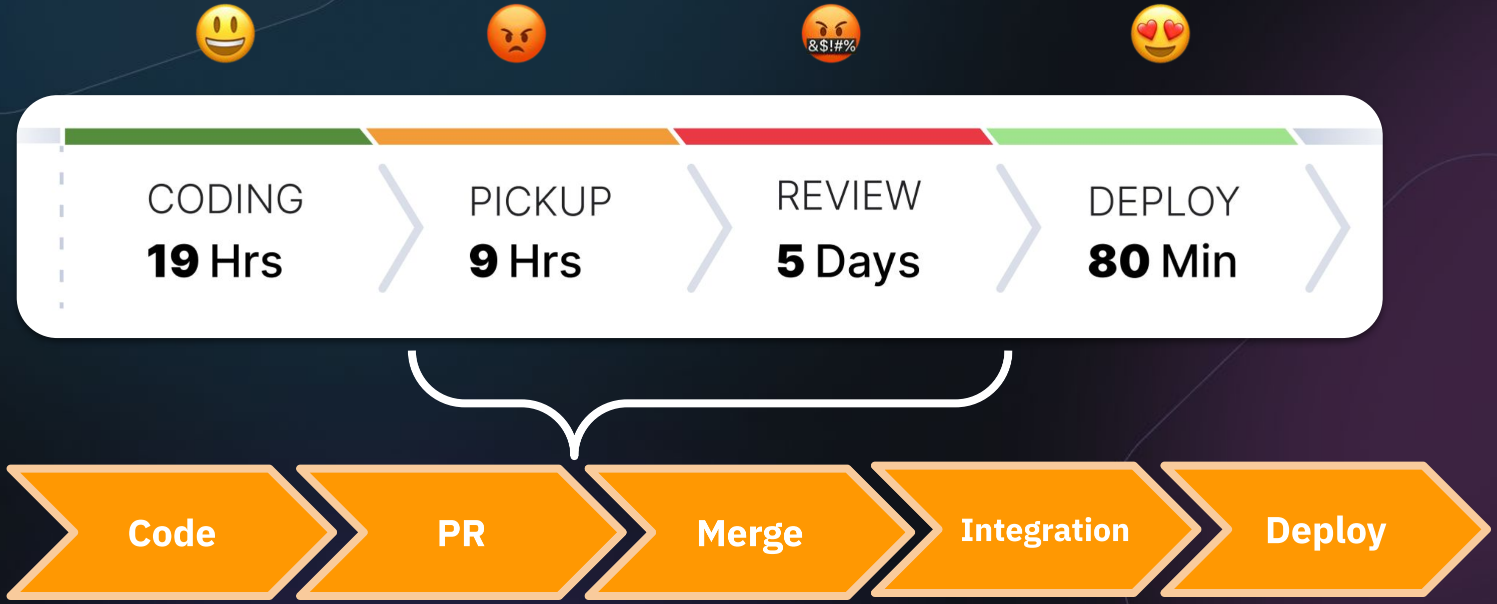
Integration

Deploy

CI
Continuous Integration

CD
Continuous Deployment

Code is getting **stuck** here



😃	😡	🤬	😍
CODING	PICKUP	REVIEW	DEPLOY
19 Hrs	**9** Hrs	**5** Days	**80** Min

Code

PR

Merge

Integration

Deploy

LinearB Labs

Pull Request Research

~1 Million	3.9M	25.6K
Pull Requests	Reviews	Developers

/:\ gitStream

Pull Requests are **mostly idle**

50% of PRs are idle
50.4% of lifespan

33% of PRs are idle
77.8% of lifespan

No Movement == No Merging

Delays cause... **more delays!**

For PRs that span more than one day:

50.1% of PRs are idle

81.3% of lifespan

Get your PR picked up **within 1 hour**
to give it a chance to be a **“Same Day PR”**

/:\ gitStream

So, what's **missing?**

Knowledge
for developers

Who should review this?

What's required to get this merged?

What's the best way to push this forward?

Context
for Reviewers

What kind of change?

Should I look at this? Now?

How much effort?

Is it safe? What's the risk?

Flexibility / Automation
for the team

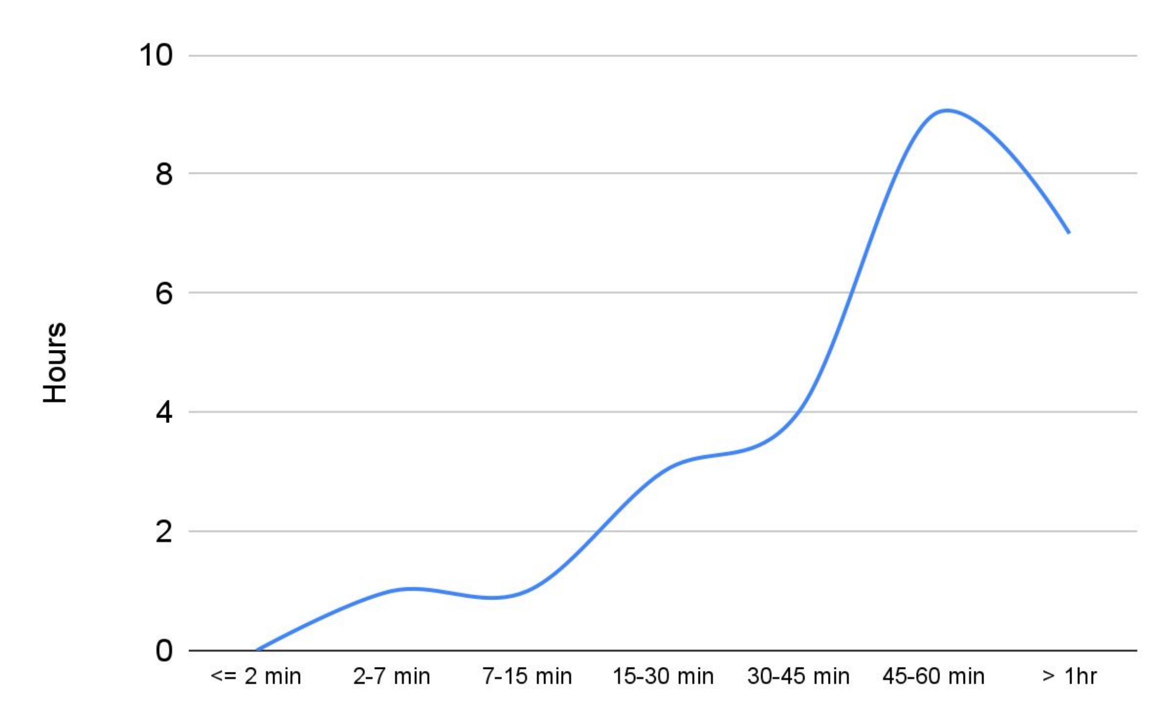
Can we reduce toil?

Must we review all PRs in the same way?

How do we enforce best practices?

/:\ gitStream

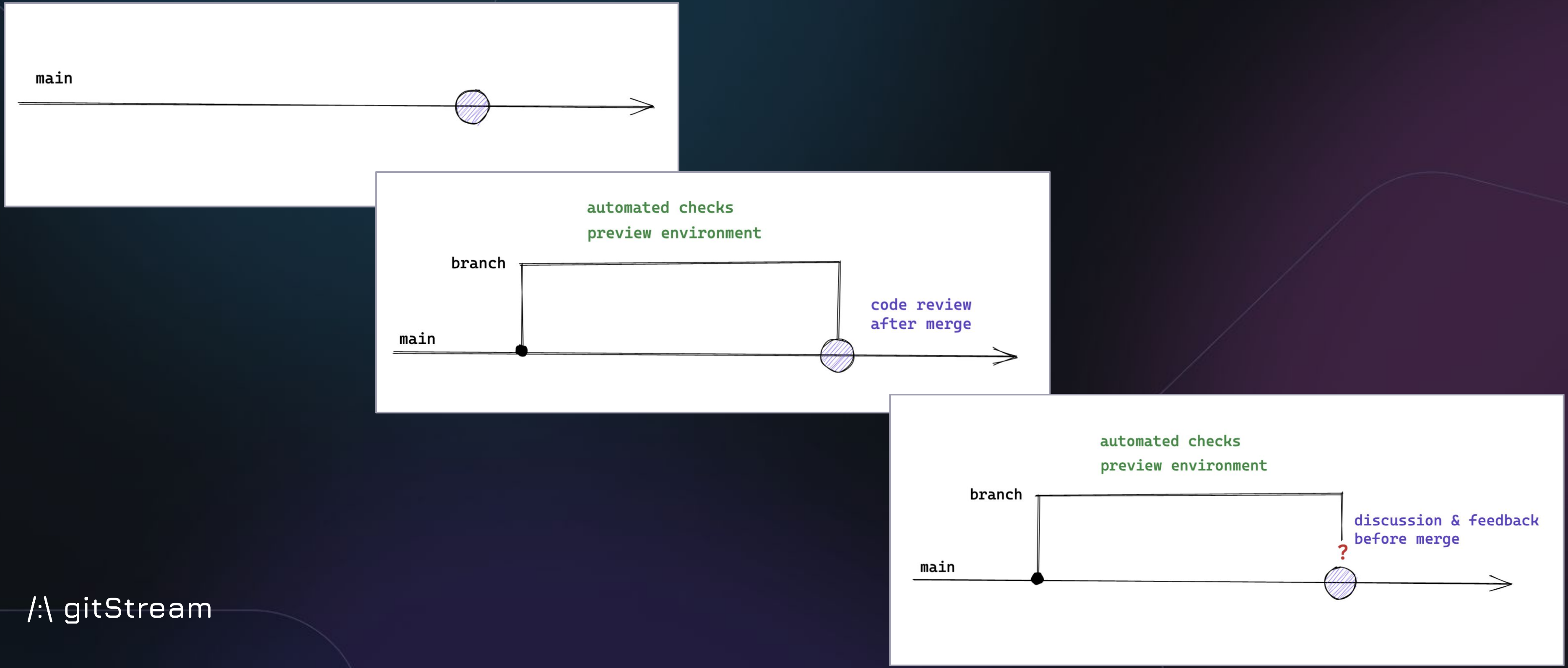
Estimated Review Time



/:\ gitStream

Ship / Show / Ask

(Rouan Wilsenach)



Thinking about a Solution

Open Discussion

Required Capabilities?

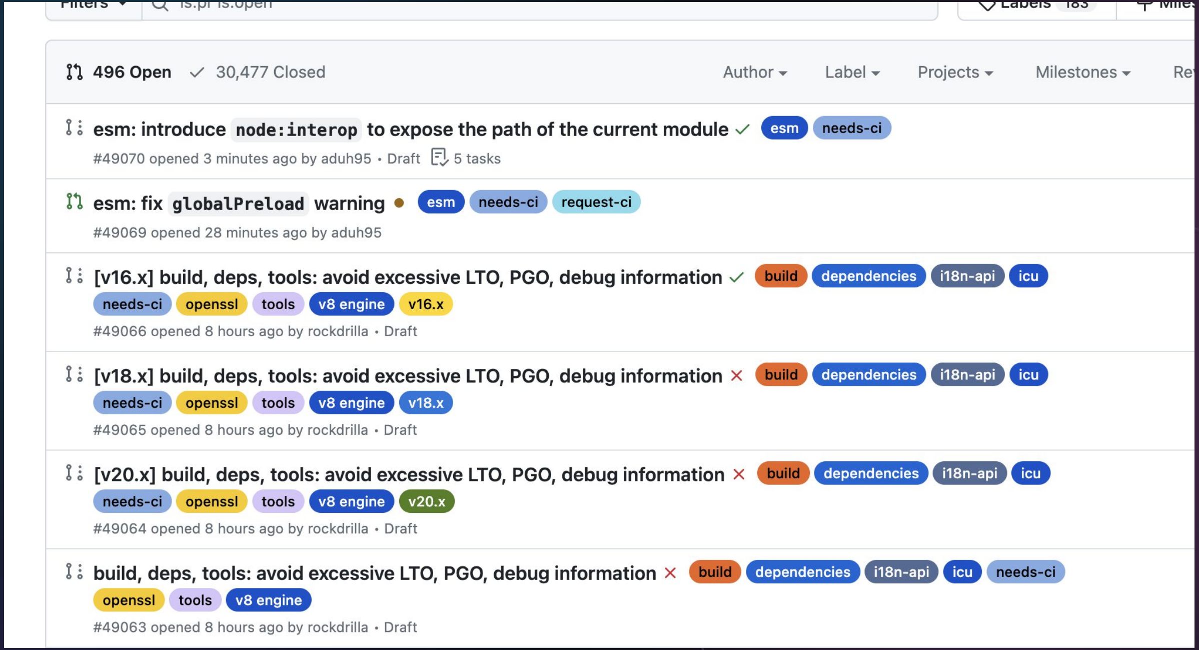
How should this run?

What to avoid?

Building Blocks for a Solution

Context

- PR Labels
- Comments
- PR title + description
- Badges



Filters ▾ is:pr is:open Labels 183 Miles

496 Open ✓ 30,477 Closed Author ▾ Label ▾ Projects ▾ Milestones ▾ Re

esm: introduce `node:interop` to expose the path of the current module ✓ esm needs-ci
#49070 opened 3 minutes ago by aduh95 • Draft 5 tasks

esm: fix `globalPreload` warning ● esm needs-ci request-ci
#49069 opened 28 minutes ago by aduh95

[v16.x] build, deps, tools: avoid excessive LTO, PGO, debug information ✓ build dependencies i18n-api icu needs-ci openssl tools v8 engine v16.x
#49066 opened 8 hours ago by rockdrilla • Draft

[v18.x] build, deps, tools: avoid excessive LTO, PGO, debug information ✕ build dependencies i18n-api icu needs-ci openssl tools v8 engine v18.x
#49065 opened 8 hours ago by rockdrilla • Draft

[v20.x] build, deps, tools: avoid excessive LTO, PGO, debug information ✕ build dependencies i18n-api icu needs-ci openssl tools v8 engine v20.x
#49064 opened 8 hours ago by rockdrilla • Draft

build, deps, tools: avoid excessive LTO, PGO, debug information ✕ build dependencies i18n-api icu needs-ci openssl tools v8 engine
#49063 opened 8 hours ago by rockdrilla • Draft

/:\ gitStream

Building Blocks for a Solution

Reviewers

- Find a (the best) reviewer
- Add a reviewer (e.g. security team; best reviewer)
  - optional / required
- Change number of required reviewers (2->1, 1->0 etc.)

Building Blocks for a Solution

Actions on PRs

- Reject (request changes) + comment
- Approve
- Merge
- Close

Advanced

Integrations

- Trigger external systems (e.g. Slack)
  - github actions!
- Play nice & leverage pipeline tools
  - Tests, Coverage, Security etc.

Scale

- Org, group and per-repo rules

Putting it all Together

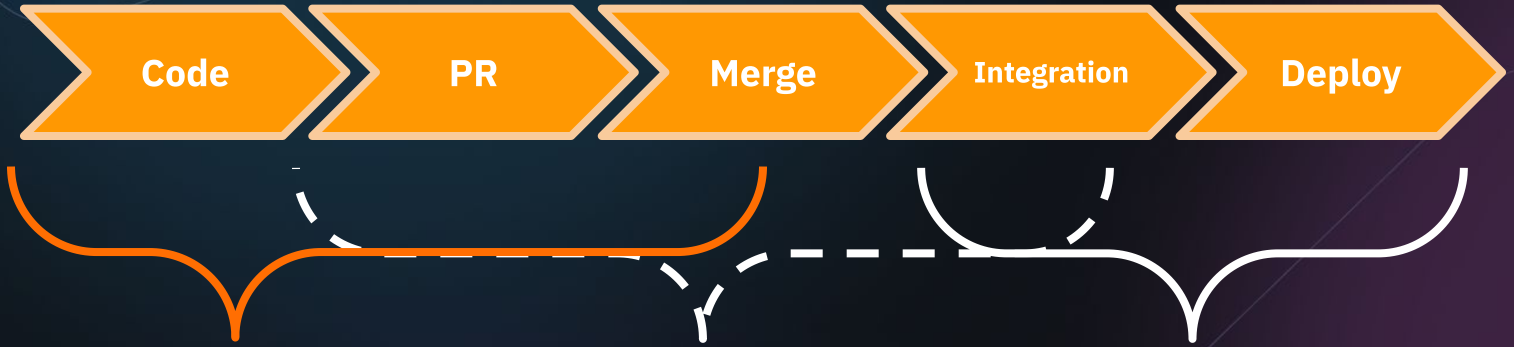
Run mode

- Run in pipeline (e.g. github actions)
- Rerun on changes in PR/MR (push, comment, etc.)

Dev Friendly - Solve our problems using code

- Programmable rule file inside the repo
    - Everyone can see the rules, understand how it works, collaborate
    - Versioned
    - Programmable logic

CM / CI / CD approach



CM
Continuous Merge

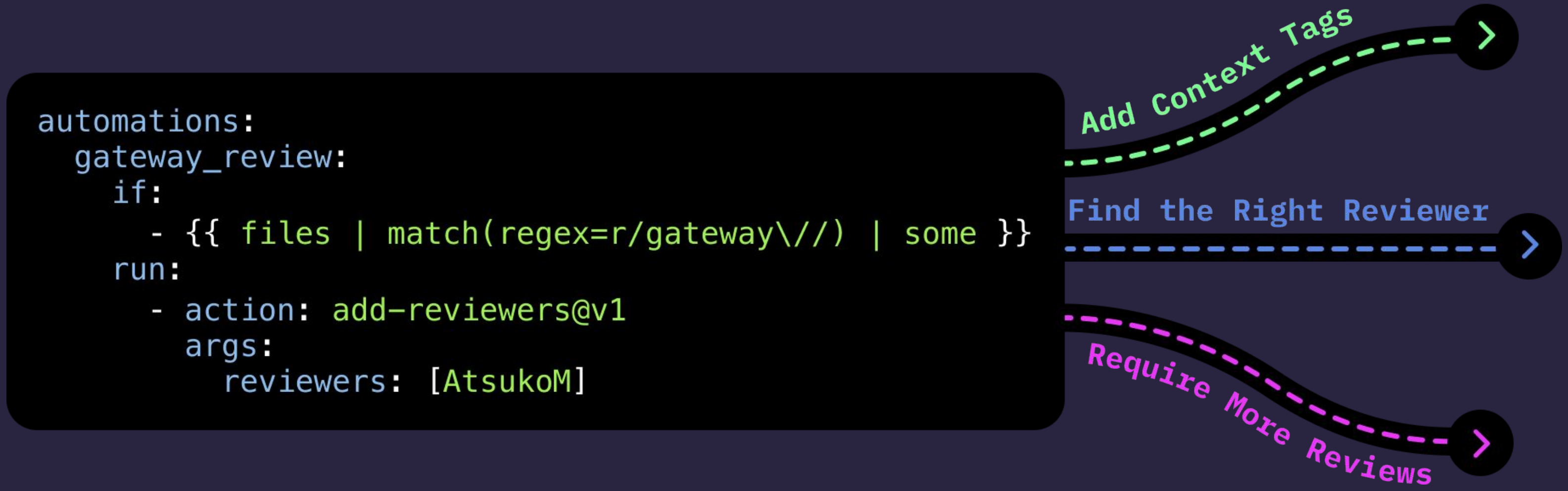
CI
Continuous Integration

CD
Continuous Deployment

/:\ gitStream

Multiple PR Review Paths

Code-as-policy to classify & route pull request



```yaml
automations:
  gateway_review:
    if:
      - {{ files | match(regex=r/gateway\//) | some }}
    run:
      - action: add-reviewers@v1
        args:
          reviewers: [AtsukoM]
```

gitStream is free!

https://gitstream.cm

Thank You!

Q&A

Learn more about
/:\ gitStream