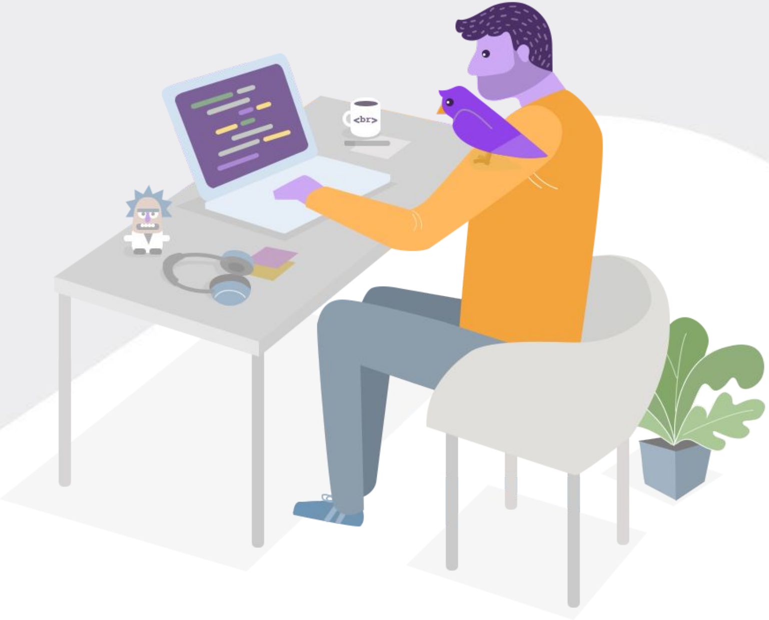
ROOKOUT

GitHub Enforcer - How Rookout enforces good dev practices on GitHub

Mickael Alliel, Rookout



Who Am I?

- DevOps @ Rookout
- Self-taught developer
- Work smarter, not harder
- Passionate about automation

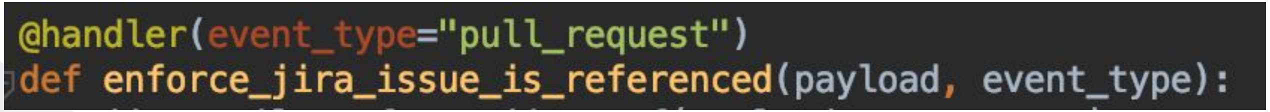
Once upon a time...

Github Enforcer

WHAT

WHY

HOW



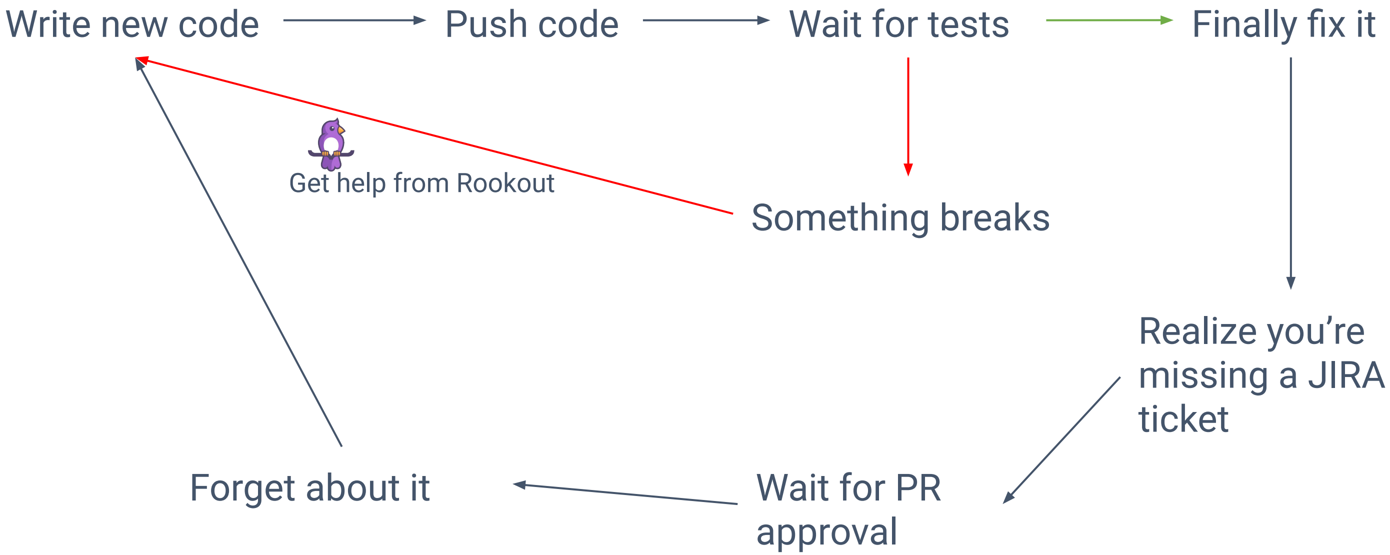
```
@handler(event_type="pull_request")
def enforce_jira_issue_is_referenced(payload, event_type):
```

What am I going to talk about ?

Github Enforcer - Yes, it’s a very generic name.

- What can it do? Coffee?
- Why do we need it?
- But.. did we really need to write our own?
- The hard moments

Typical day in the skin of a developer



Write new code

Push code

Wait for tests

Finally fix it

Something breaks

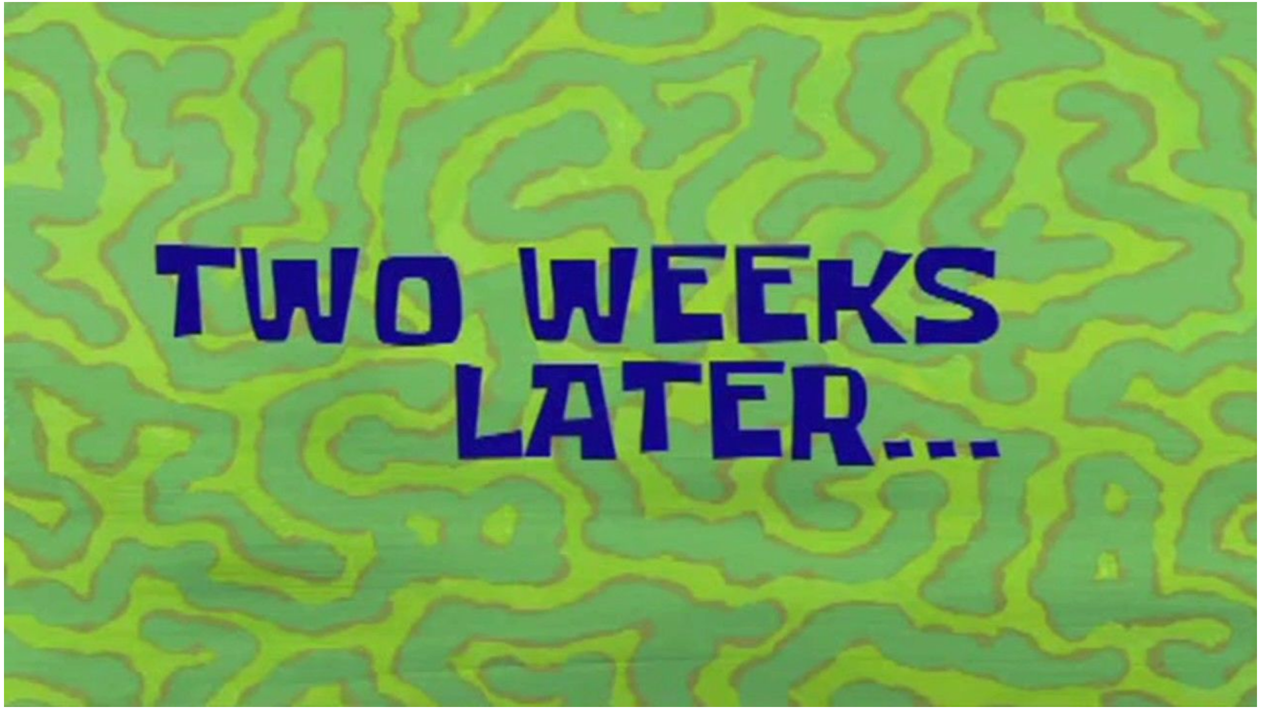
Get help from Rookout

Realize you're missing a JIRA ticket

Wait for PR approval

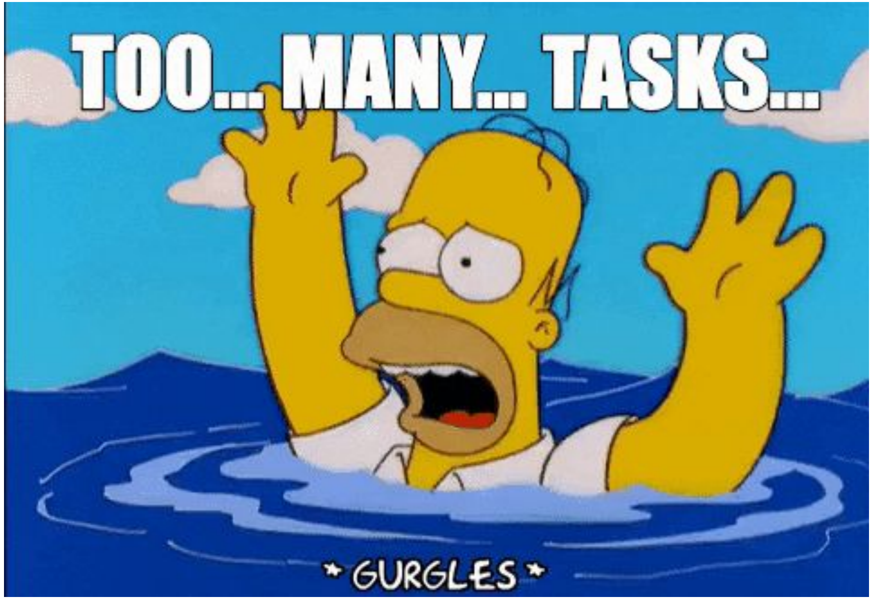
Forget about it

Typical day in the skin of a developer

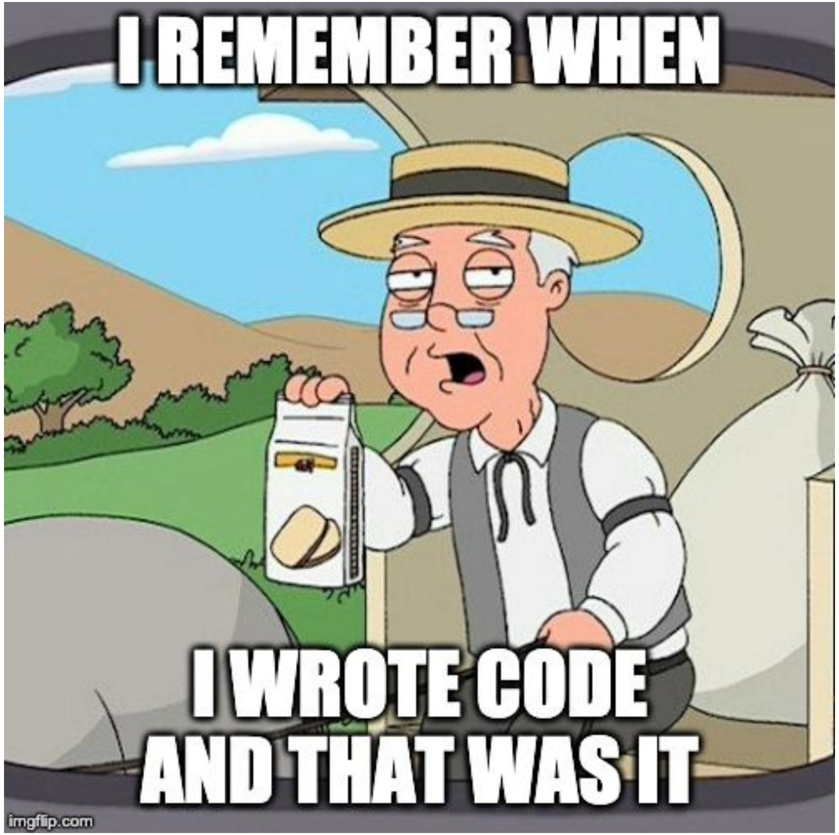


Typical day in the skin of a developer

Your code is now in staging.
Good job !



Developer pains



Developer pains

Track your PRs

Manage your kanban

Work fast

Deal with CI/CD issues

Developer pains

Dealing with life - Automating your workflow

Automatic Merge

Notifications on old stale PRs

Update with master branch

Dealing with life - Automating important bullsh**

Automatic ticket state transitioning

Release notes

Labels enforcement

Various QoL improvements

Dealing with life - Automating actually important stuff

Slack notifications on success/failures

Dry-run release notes

3rd party integrations sync

Opening tickets `/jira open task`

Ok, but why did we need all of that?

Bla bla bla, automation is trending

Bla bla bla, better best practices for development

Bla bla bla, we need pipelines - release notes, audit, labels, checks

Ok, but why did we need all of that?

As I said -

BLA BLA BLA

Ok, but why did we REALLY need all of that?

TO SAVE TIME !

Isn’t there already something out there?

- At the time - there were nothing that could do ALL of it.
- Choosing to add a service to your company’s flow is already tough enough.
- Creating your own makes it easier to customize and match your workflow, instead of customizing your workflow to match your automation.
- Today is easier to use something already made (Probot Apps, Github Apps/Actions)

Wallowing in despair

- Understanding GitHub’s API
- Rate-limiting
- Concurrency
- Regrets about Python
- Testability & Development process for a GitHub service

Should you also build your own?

PRO	CONS
Save time in the long run	It’s not a 1 day project to build
Makes your life a LOT easier	Tests are hard
It’s cool - look, I’m talking here	Things break - and you will get the blame

Alternatives? Probot, GitHub Actions, Serverless

GitHub apps scraps? Mergify, Issue-Label Bot, Stale

Thank you !

Questions ?

rookout.com

ROOKOUT