



Small Changes, Big Impact: GitHub Workflows for the LLVM Project

David Spickett, Arm / Linaro

FYI

- This is quite high level.
- There will be more text than you can read.
- I will talk about the most important bits.
- Come back to the slides later if you find this interesting.

GitHub Actions Workflows

- Automate tasks for a GitHub repository
- “workflows” define what to do and when
- Run on remote “Runners” provided by GitHub
- Workflows defined in YAML
- [.github/workflows](#) in llvm-project



Actions

← Files	main ▾	llvm-project / .github / workflows /
build-ci-container.yml	[Github] Bump CI container to ...	
build-metrics-container.yml	[Github] Prefer ubuntu-24.04 o...	
ci-post-commit-analyzer-ru...	[workflows] Add post-commit jo...	

LLVM Workflows

If you have contributed to LLVM, you have seen a workflow in action.

- Pull Request (PR) Labelling
- Checking code formatting
- Backporting to releases
- ...more examples later...

Actions

New workflow

All workflows

Build and Test libc++

Build CI Container

HLSL Tests

Add buildbot information t...

AI Code Reviewer

Build Docker images for li...

Build Metrics Container

Build Windows CI Contai...

Check code formatting

Check for private emails ...

Check libc++ generated fi...

CI Tests

YAML Example

What can it access? →

```
permissions:
  contents: read
```

When should it run? →

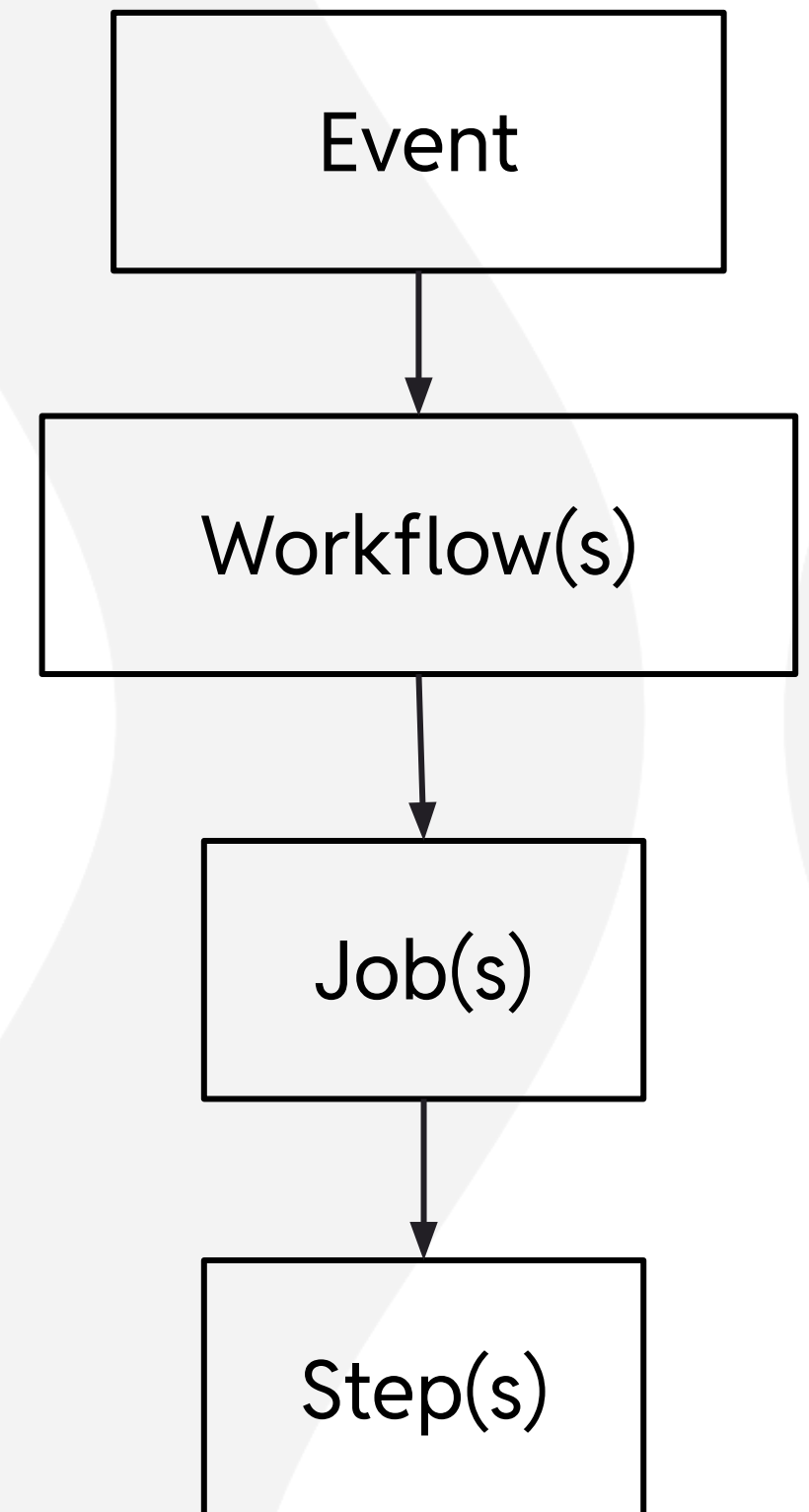
```
on:
  push:
```

What should it run? →

```
jobs:
  check-docs-build:
    name: "Test documentation build"
    runs-on: ubuntu-24.04
    steps:
      - name: Fetch LLVM sources
```

Where should it run? →

What work should it do? →



So You Want to Write a Workflow

You need:

- A Github account
- Patience

No special membership required, even the Runners are free to everyone*.

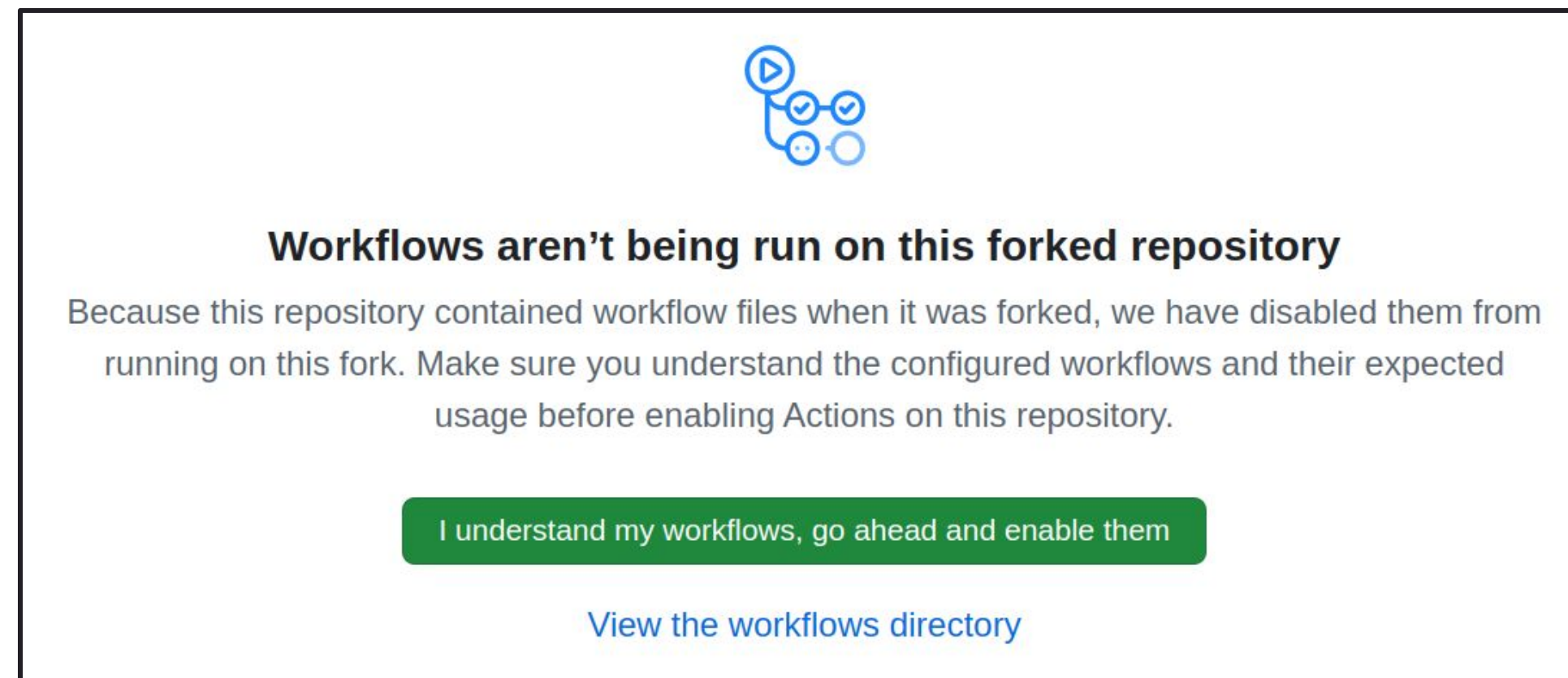
* <https://docs.github.com/en/billing/managing-billing-for-your-products/managing-billing-for-github-actions/about-billing-for-github-actions#included-storage-and-minutes>

Fork LLVM



Workflows and Runners will be available by default.

You will need enable the existing workflows from the “Actions” tab:



LLVM's workflows have repository name checks as well, more on this later.

Starting Points

- YAML is tricky, GitHub's errors make it worse.
- Copy an existing workflow.

Look for similar:

- Trigger Events (push, new PR, new issue, ...)
- Use Case (checking code, managing labels, inspecting builds, ...)
- Results (writing a comment, creating a file, adding a label, ...)

For example...

New PR Workflow - new-prs.yml

- Runs when a new PR is opened.

on:

pull_request_target:

types:

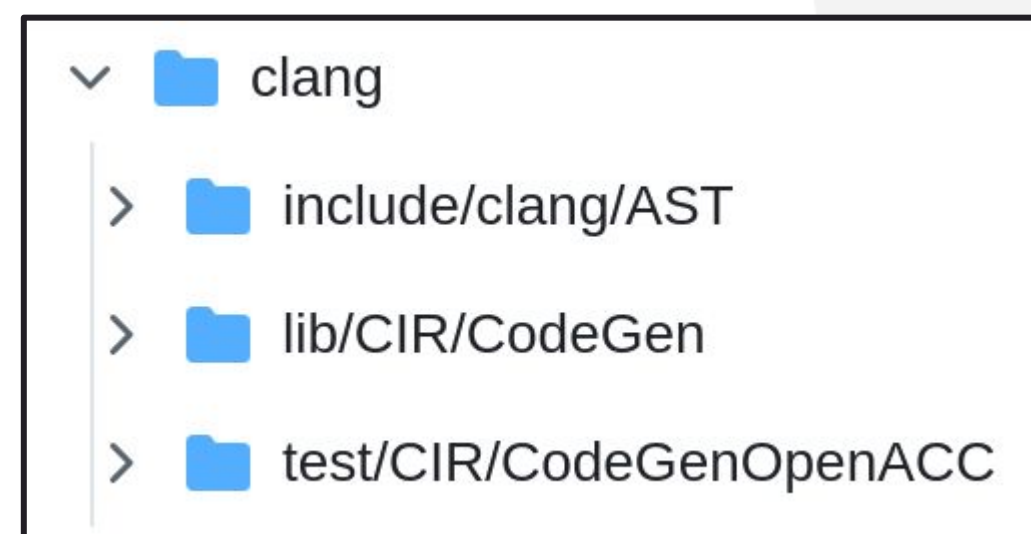
- opened
- reopened

<...>

- Labels it based on the changes.

steps:

- uses: actions/labeler



LlvmBot added **clang** **clang:frontend**

ClangIR labels 10 hours ago

New PR Workflow - new-prs.yml

- Greet new contributors.



- name: Greet Author

run: |

```
python3 ./github-automation.py <...> pr-greeter <...>
```

github-automation.py - a collection of uses of the GitHub API.

- Specific to LLVM
- Built on [PyGitHub](#)

Enable On Your Fork

- Workflow files may have extra checks:

```
if: github.repository == 'llvm/llvm-project'
```

- Change to your username, change back when submitting to LLVM.

```
if: github.repository == 'your-username/llvm-project'
```

(you only need to do this for the workflow you are editing)

(and no, the UI [does not tell](#) you [why](#) it was skipped, that would be far too useful)



This job was skipped

Write the rest of the workflow

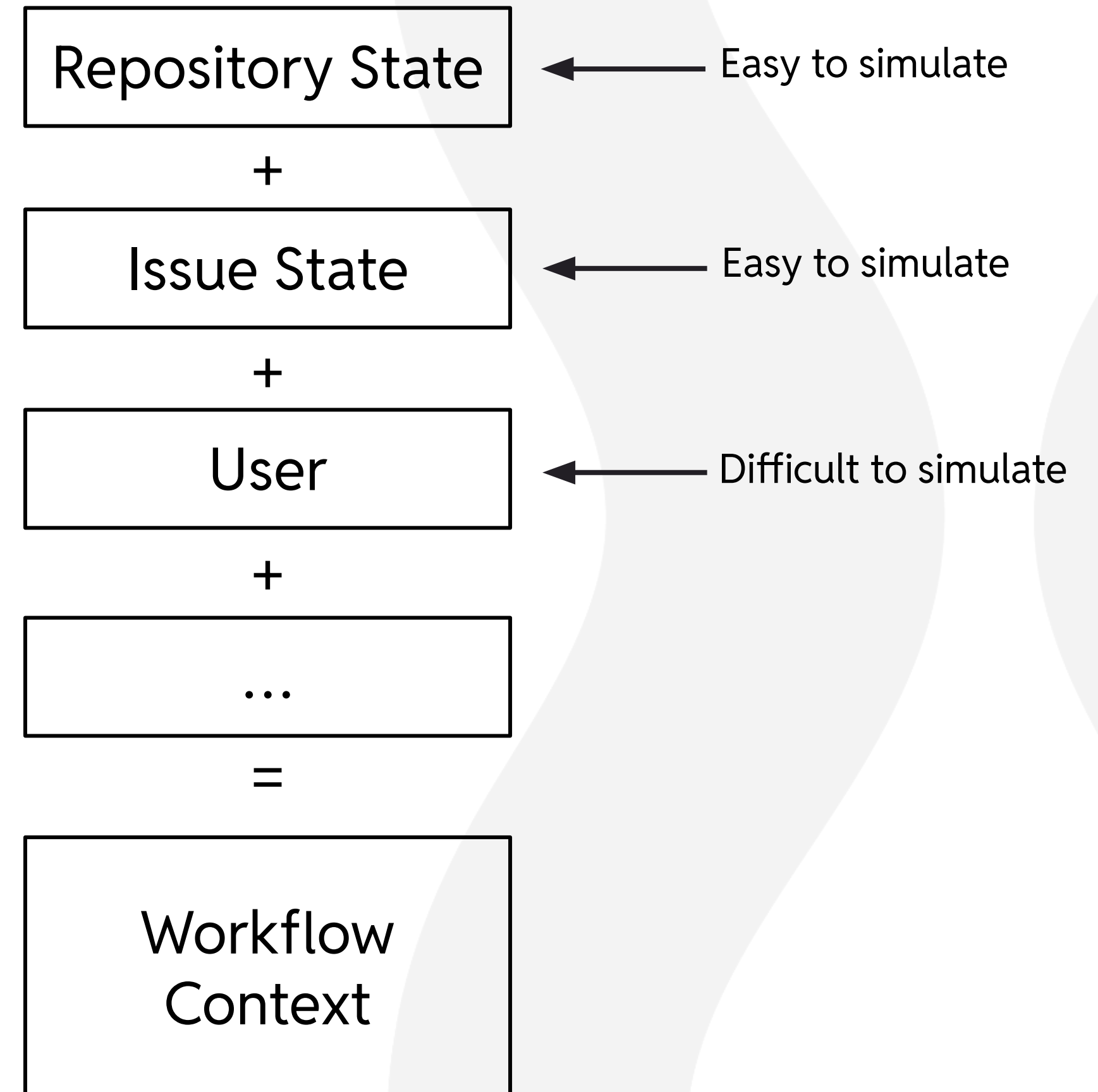
- Use a YAML aware editor.
- Edit in small chunks and commit often.
- Put complex logic into script steps or script files, instead of YAML.*
- Use the GitHub [documentation](#).
- GitHub Code Search to find examples.†

* Unless doing it in YAML saves a lot of compute time.

† You might be the first of 100M users to notice that something is [broken](#).

Testing

- GitHub Runners are not under our control.
- No local testing.
- No test framework.
- Cannot simulate all conditions.



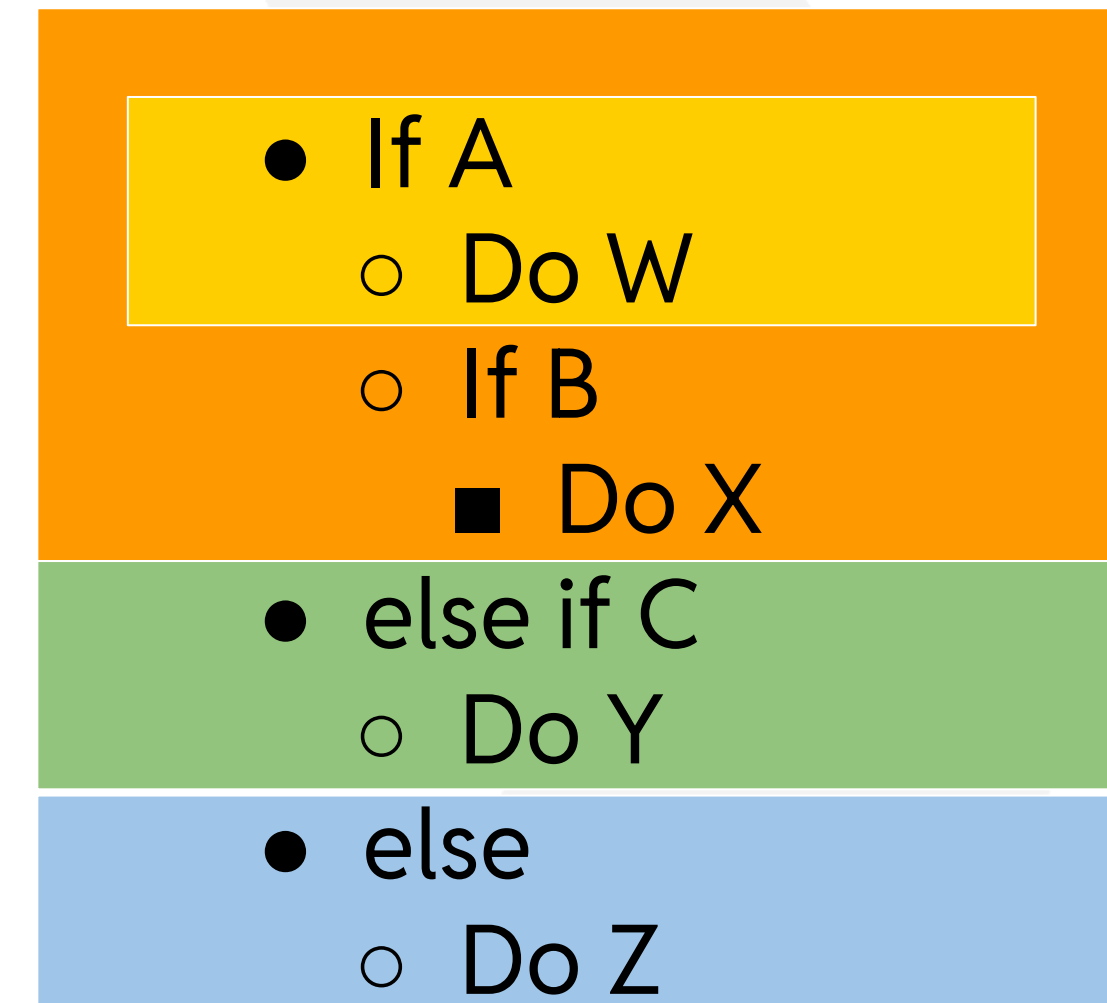
Testing

Solution:

- Manual testing.
- Cannot change all inputs but -
- Can temporarily change the code, **as if** we had changed the inputs.

Testing - Scenarios

- Write out all the scenarios that you need to handle
- Split scenarios into parts if needed
- Make changes to simulate one part
- Test
- Reset changes
- Repeat for all parts



(each colour is one test run)

Testing - New PR Example

Scenario 1: New PR from a new user

Expected: Labels and greeting comment added

Test with:

```
if: >-
  (github.repository == 'you/llvm-project') &&
  (github.event.action == 'opened')
```

- No check (as if user is new)
- **Do not** contribute this.

Scenario 2: New PR from an existing user

Expected: Labels added

Test with:

```
if: >-
  (github.repository == 'you/llvm-project') &&
  (github.event.action == 'opened') &&
  (github.event.pull_request.author_association !=
  'COLLABORATOR') &&
  <...>
```

- Does the check.
- **Do** contribute this.

Scenario 1 + Scenario 2 = Full Coverage*

* As much as we can hope for given we are not “testing what we ship”

Contributing to LLVM

- Save a copy of the branch, in case you need the history.
- Remove testing hacks, skips and manual triggers.
- Reset “repo==” checks to “**llvm**/llvm-project”.
- Squash into one commit.
- [Send a PR](#) to LLVM as you would for any other change.

Labels

github:workflow

Ideas

Existing

- Labelling PRs
- Easy backports
- Formatting checks
- Managing commit access

In progress*

- Pre-merge testing⁺
- Clean up user branches
- Find someone to merge a PR for you

Future?

- Per-file [reviewer notes](#)
- Unusual builds on demand
- Recommended tests
- Domain specific linters
- ...

* As of 2025-02-04

⁺ Pre-merge testing is moving from Buildkite to GitHub

Conclusions

- You all have the ability to edit and create your own workflows!
- Make life easier for 100s of contributors every day.



linaro.org

Thank you

“Writing a new workflow” workflow

- Fork llvm-project
- Copy a YAML file with a similar use case
- Enable the workflow on your fork
- Write the rest of the workflow
- Push to your fork’s main branch
- Trigger the Workflow
- See the result in the “Actions” tab
- Fix mistakes, push, test, repeat until working.
- Squash
- Send PR to LLVM