UNIVERSITÄT
BERN

Exploring GitHub Actions through EGAD: An Experience Report

Pablo **Valenzuela**, Alexandre **Bergel**, Timo **Kehrer**, Oscar **Nierstrasz**

SEG @ University of Bern

GitHub Actions

- Support the **automation** of **Software Engineering tasks**
- **Increasingly adopted** in software repositories
- For **example**: Notifications, Pull request/Issue messages, Continuous Integration
- The most **widely used tool** for Continuous Integration

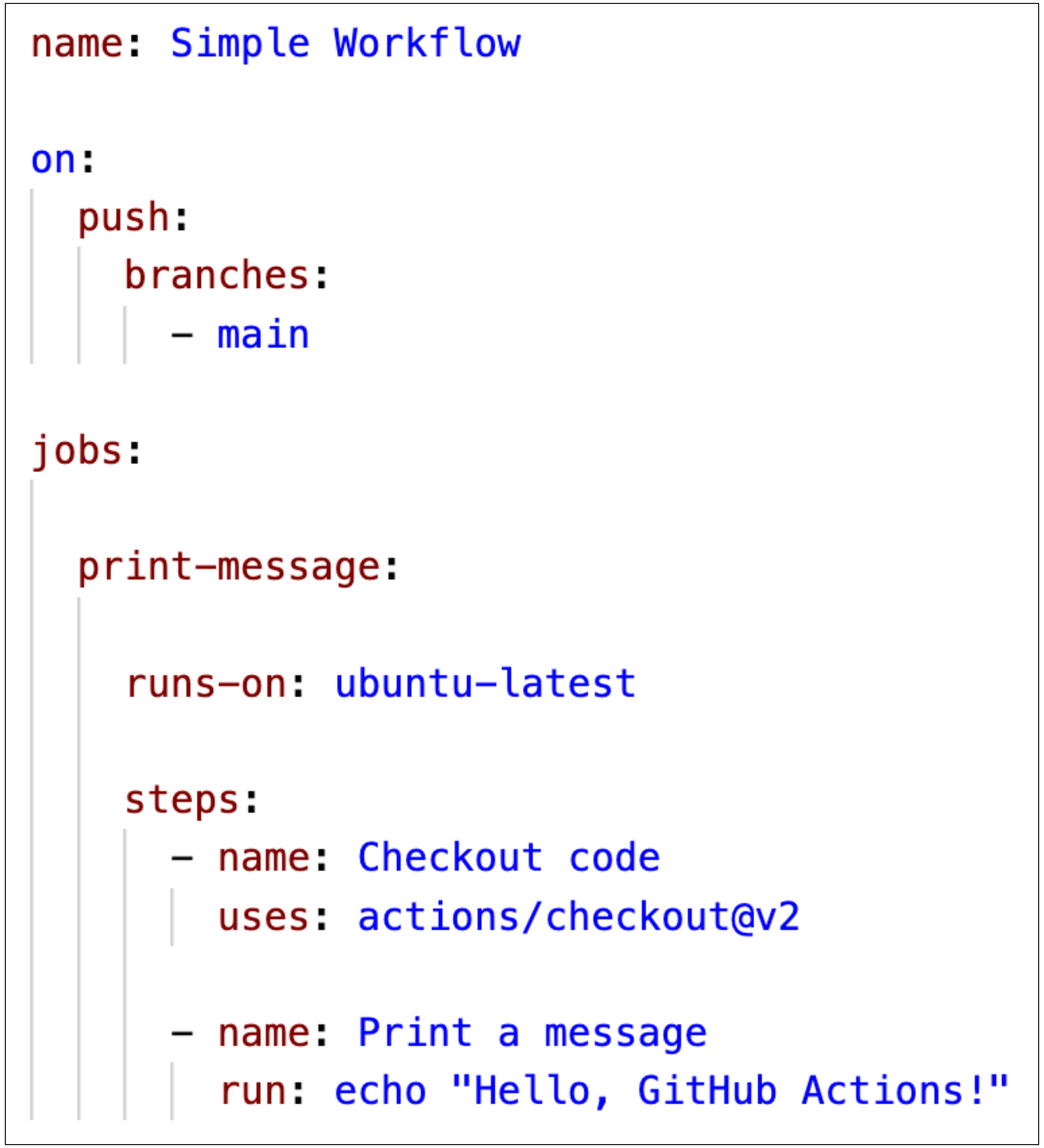
How GitHub Actions automate software engineering tasks?

Workflow YAML file:

- Components
- Execution
- Specification process

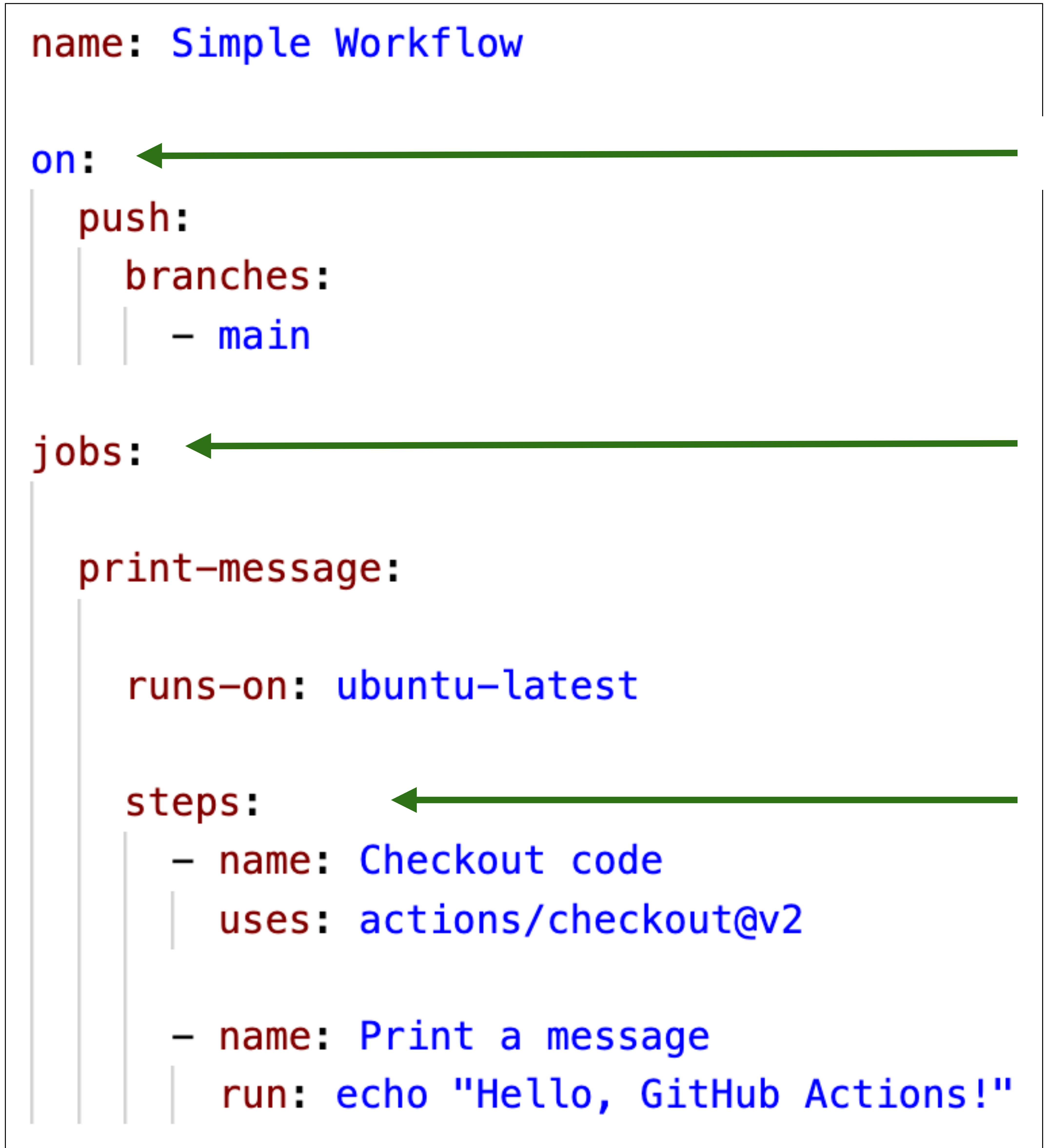
Workflow

The workflow YAML file



```yaml
name: Simple Workflow

on:
  push:
    branches:
      - main

jobs:

  print-message:

    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Print a message
        run: echo "Hello, GitHub Actions!"
```

The workflow YAML file



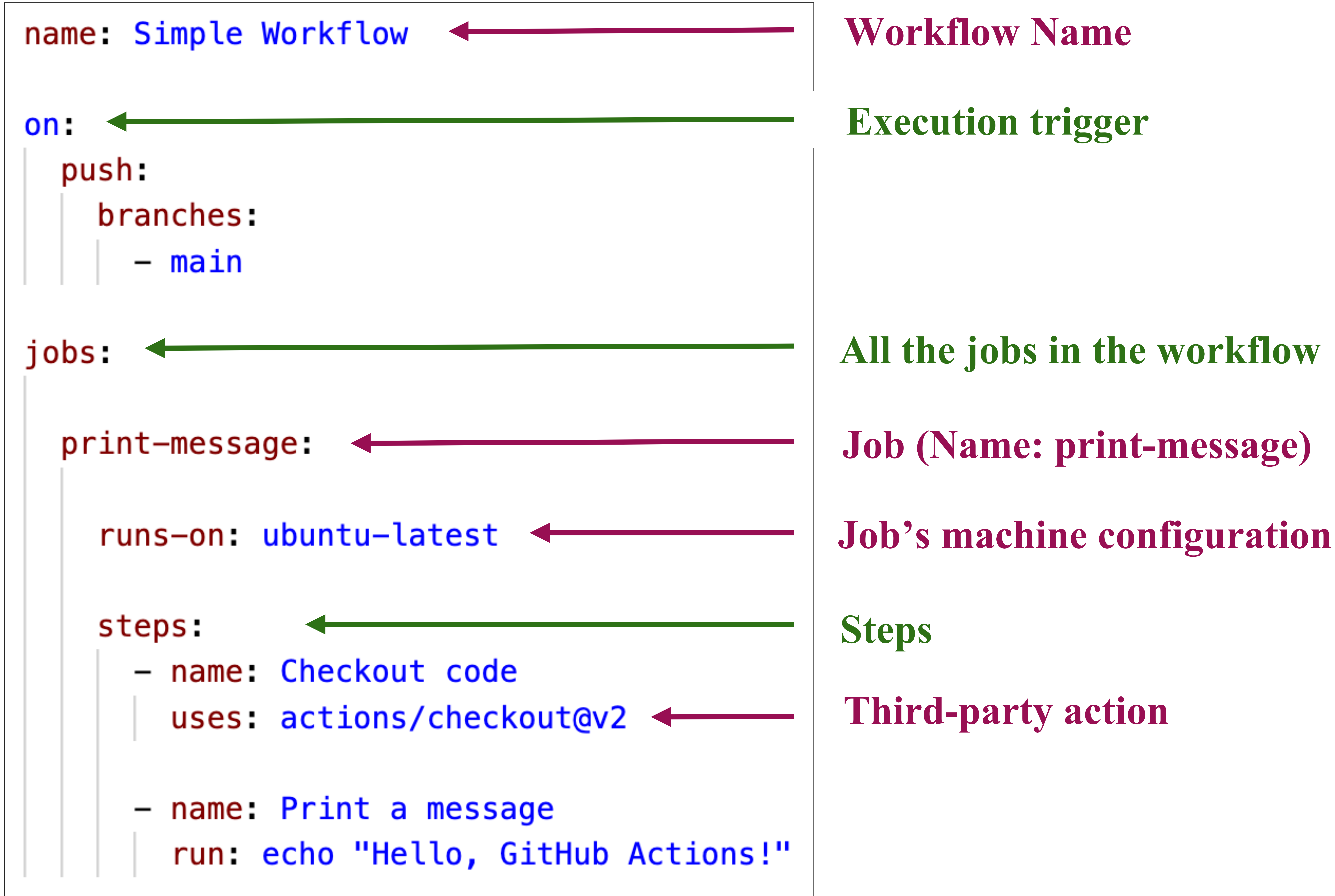
```yaml
name: Simple Workflow

on:
  push:
    branches:
      - main

jobs:

  print-message:

    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Print a message
        run: echo "Hello, GitHub Actions!"
```

Execution trigger

All the jobs in the workflow

Steps

The workflow YAML file



```yaml
name: Simple Workflow

on:
  push:
    branches:
      - main

jobs:

  print-message:

    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Print a message
        run: echo "Hello, GitHub Actions!"
```

Workflow Name

Execution trigger

All the jobs in the workflow

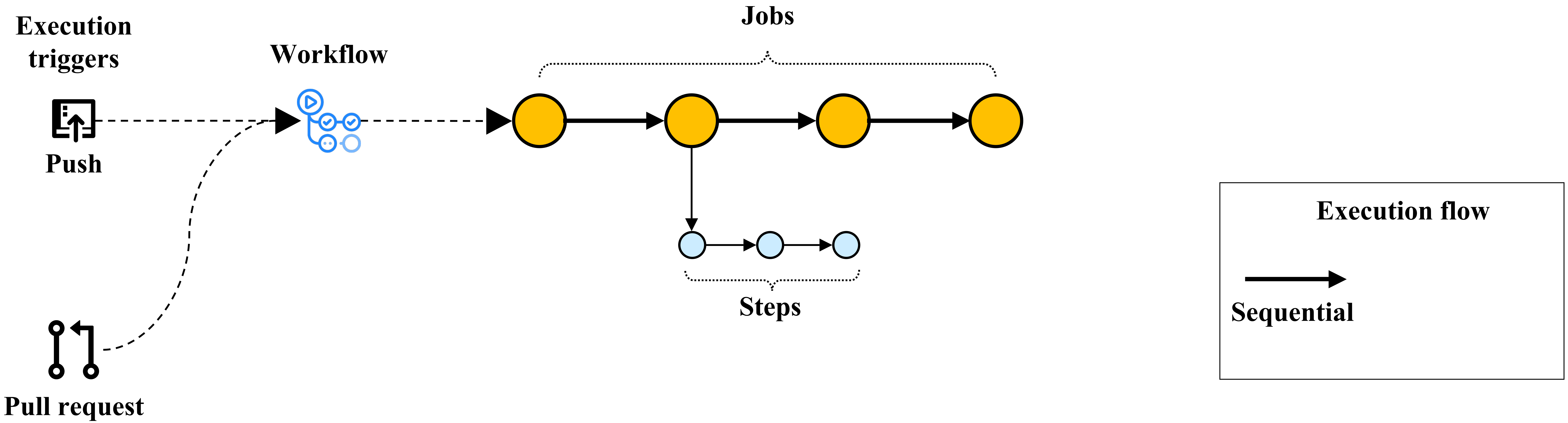
Job (Name: print-message)

Job’s machine configuration

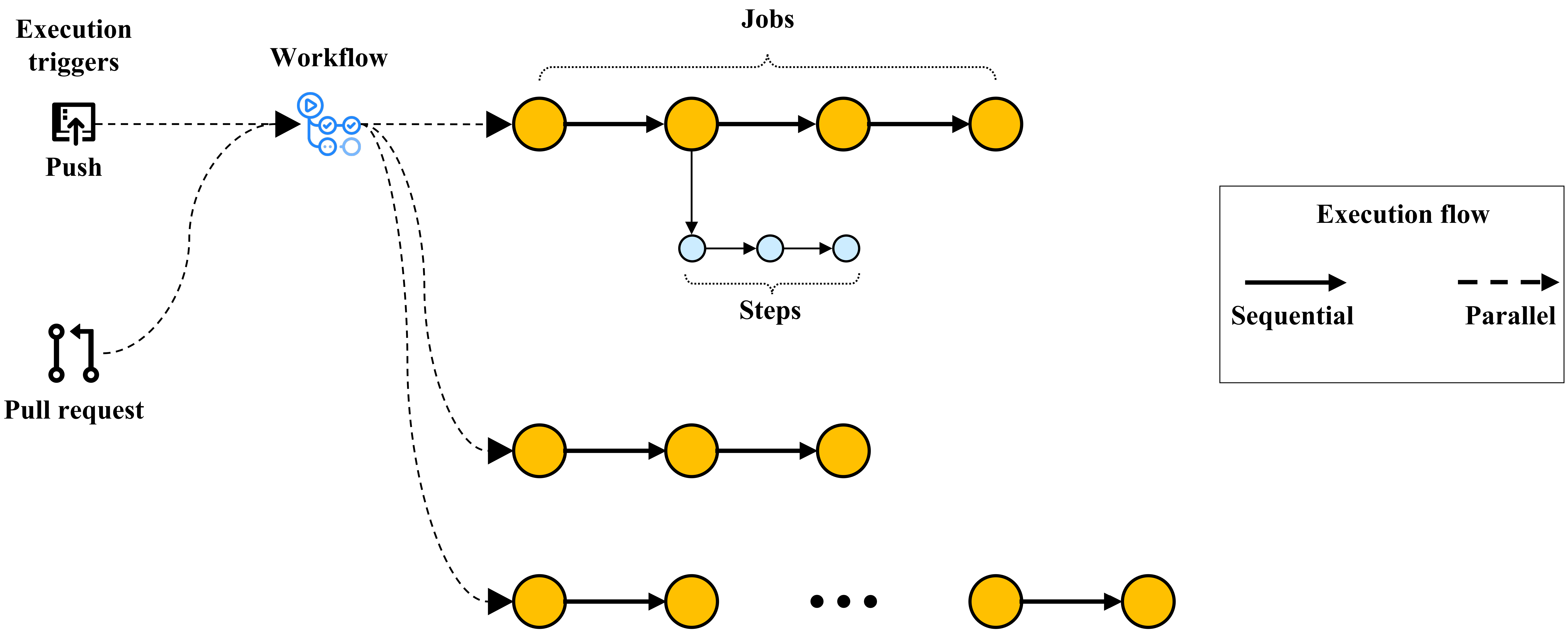
Steps

Third-party action

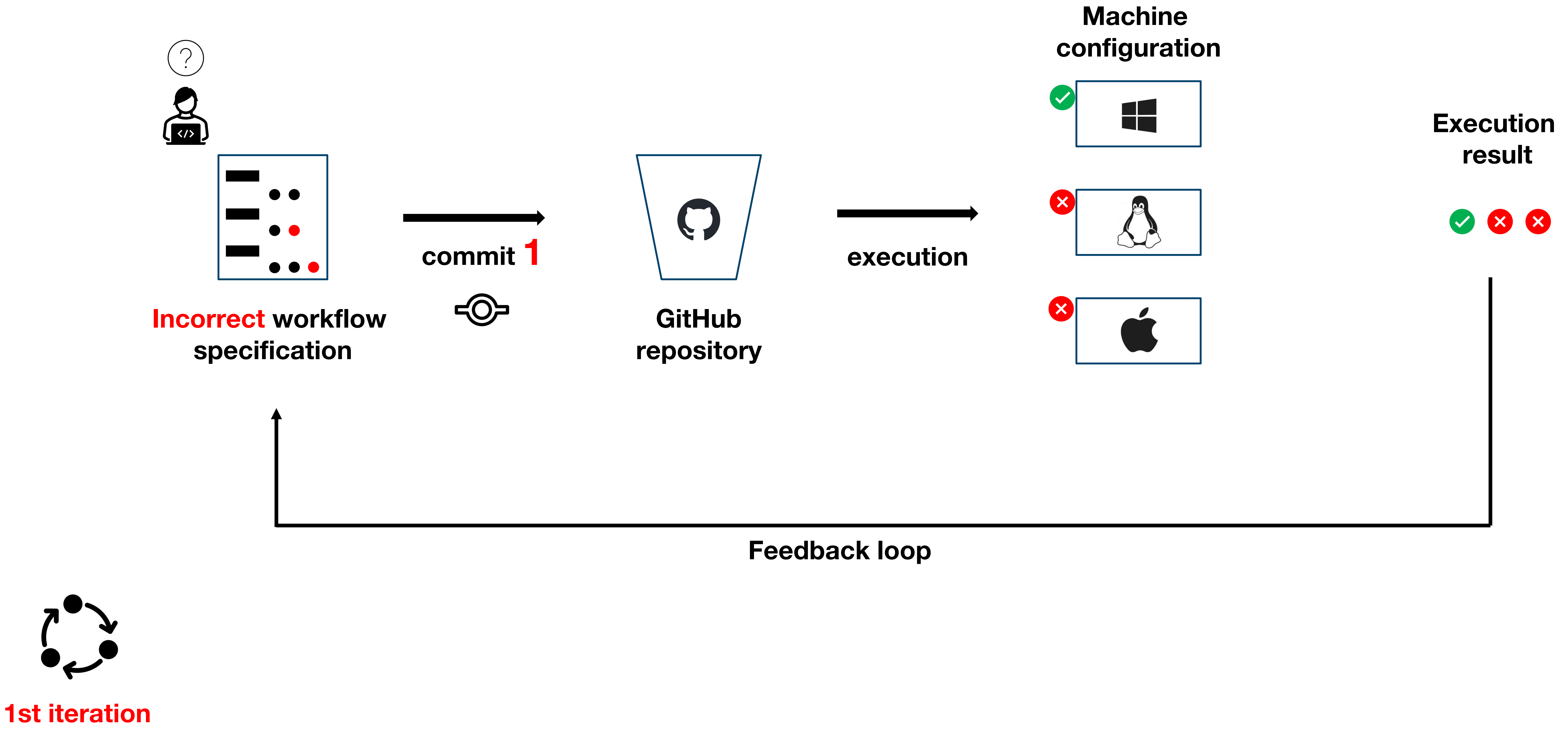
The workflow execution



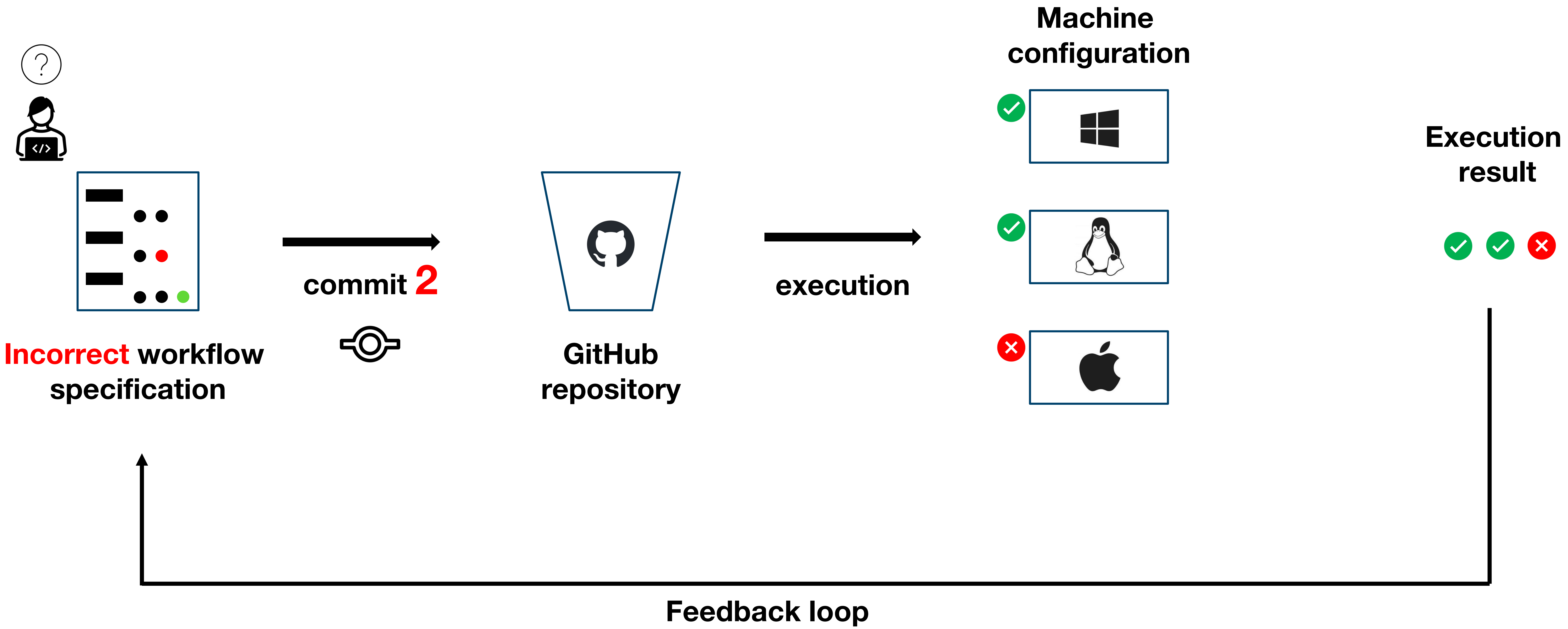
The workflow execution



Workflow specification process



Workflow specification process



Machine configuration

Execution result

Incorrect workflow specification

commit 2

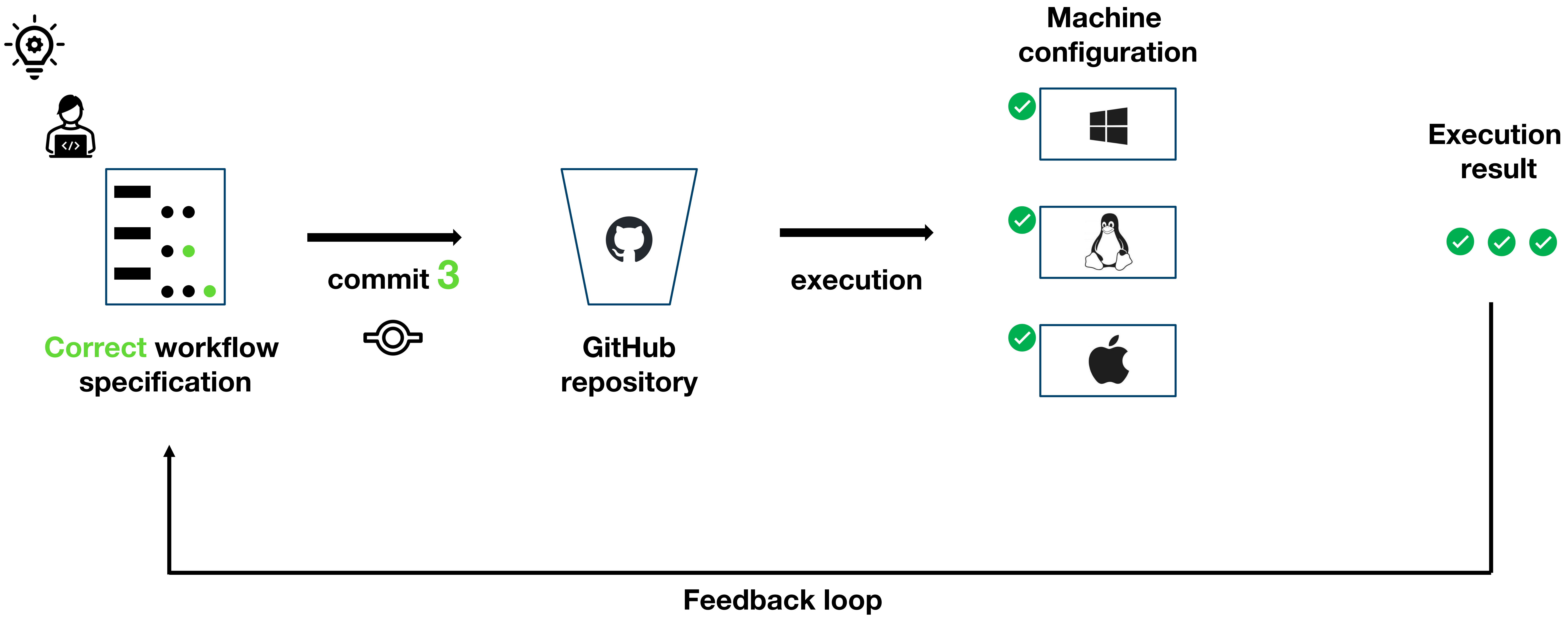
GitHub repository

execution

Feedback loop

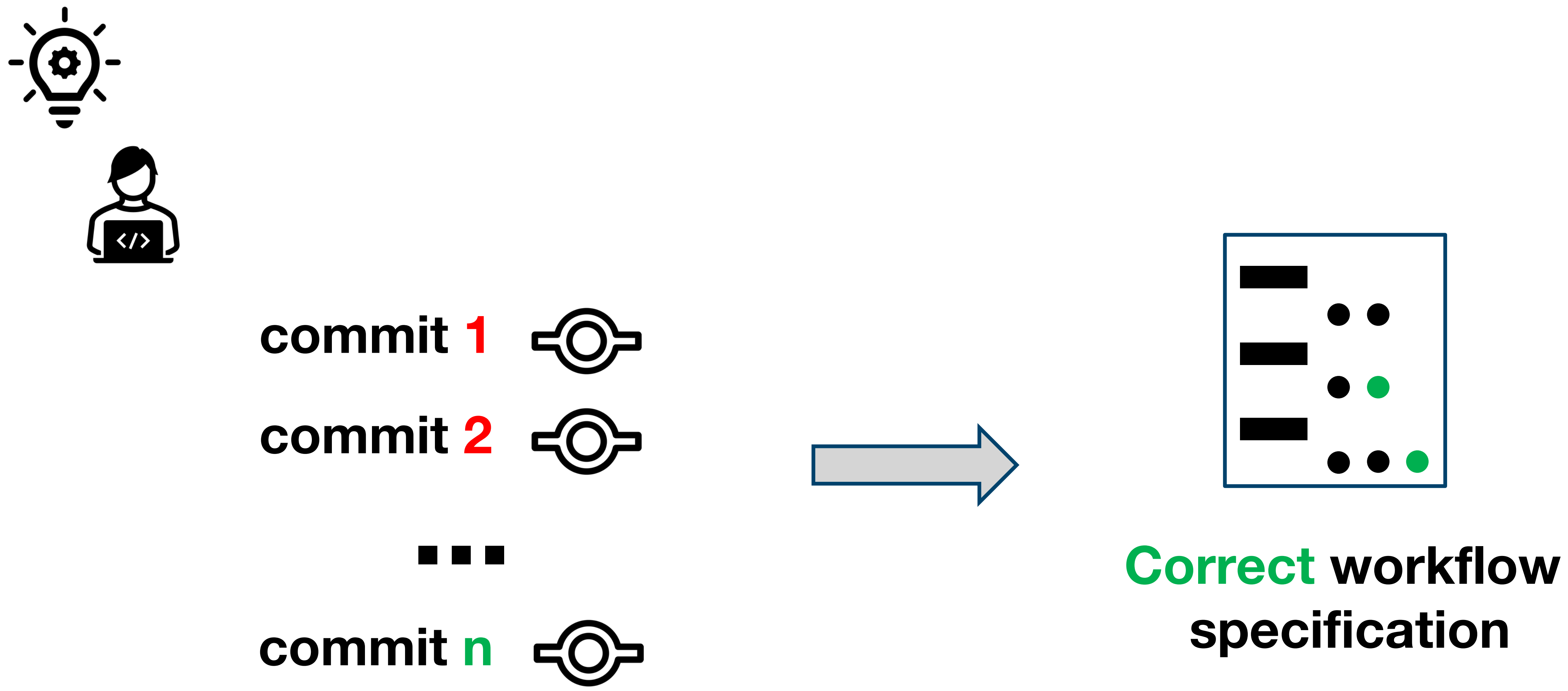
2nd iteration

Workflow specification process



3rd iteration

Workflow specification process



commit 1

commit 2

...

commit n

Correct workflow specification

Consequences:

1. Pushing multiple commits contaminate the repository's history
2. Slow feedback loop

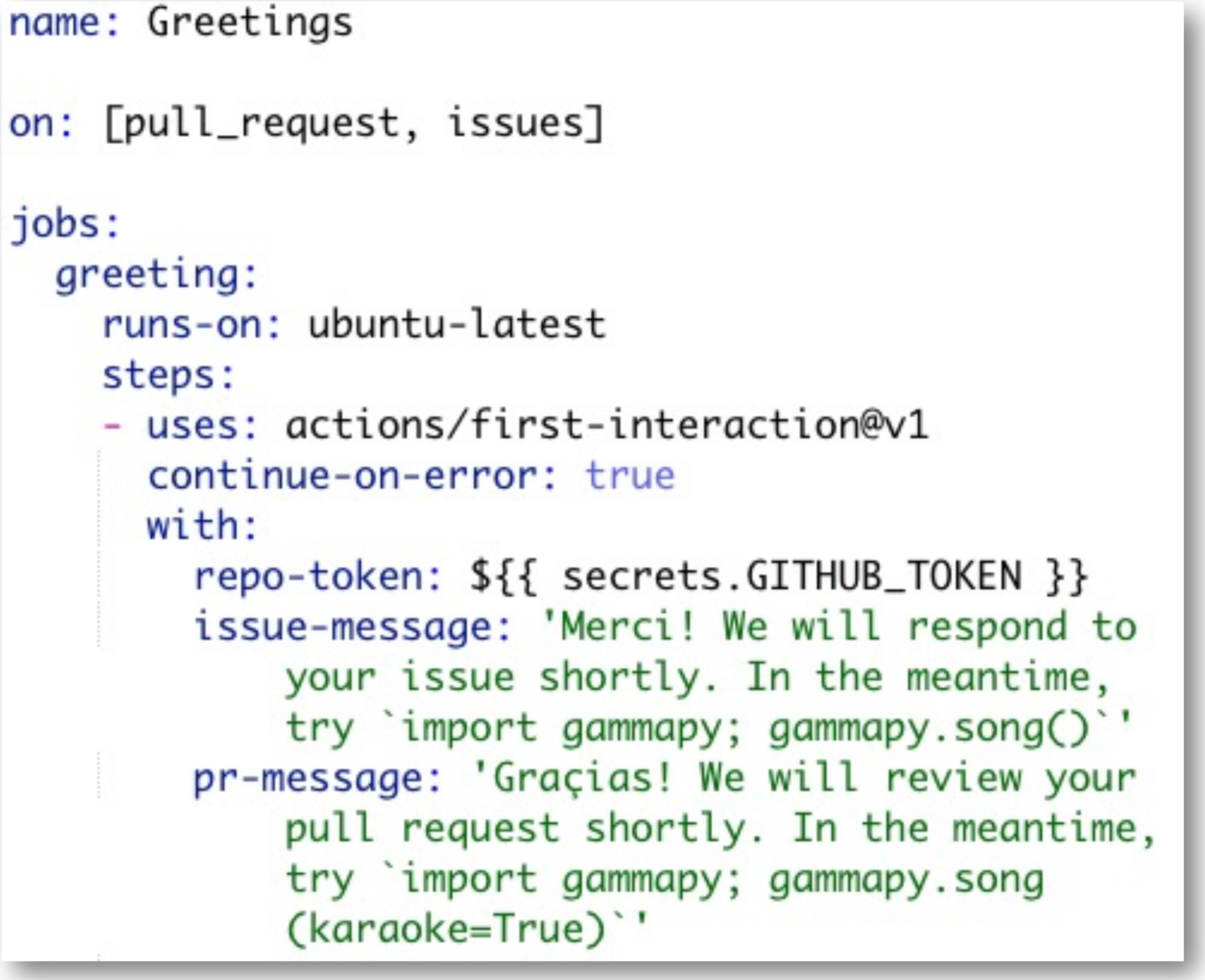
Exploring GitHub Actions through EGAD

- **EGAD**: A Moldable tool for GitHub Action Analysis
- **Learned lessons**:
    1. Compose a story,
    2. Navigating custom views,
    3. Supporting the onboarding of researchers on Glamorous Toolkit.

The need for a GitHub Action domain model

- Workflows files → **Valuable Information**
- Batch tools for GA research

1. Ignore the context
2. No domain model



```
name: Greetings

on: [pull_request, issues]

jobs:
  greeting:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/first-interaction@v1
      continue-on-error: true
      with:
        repo-token: ${{ secrets.GITHUB_TOKEN }}
        issue-message: 'Merci! We will respond to
            your issue shortly. In the meantime,
            try `import gammapy; gammapy.song()`'
        pr-message: 'Graçias! We will review your
            pull request shortly. In the meantime,
            try `import gammapy; gammapy.song
            (karaoke=True)`'
```

https://github.com/gammapy/gammapy/blob/main/.github/workflows/greetings.yml

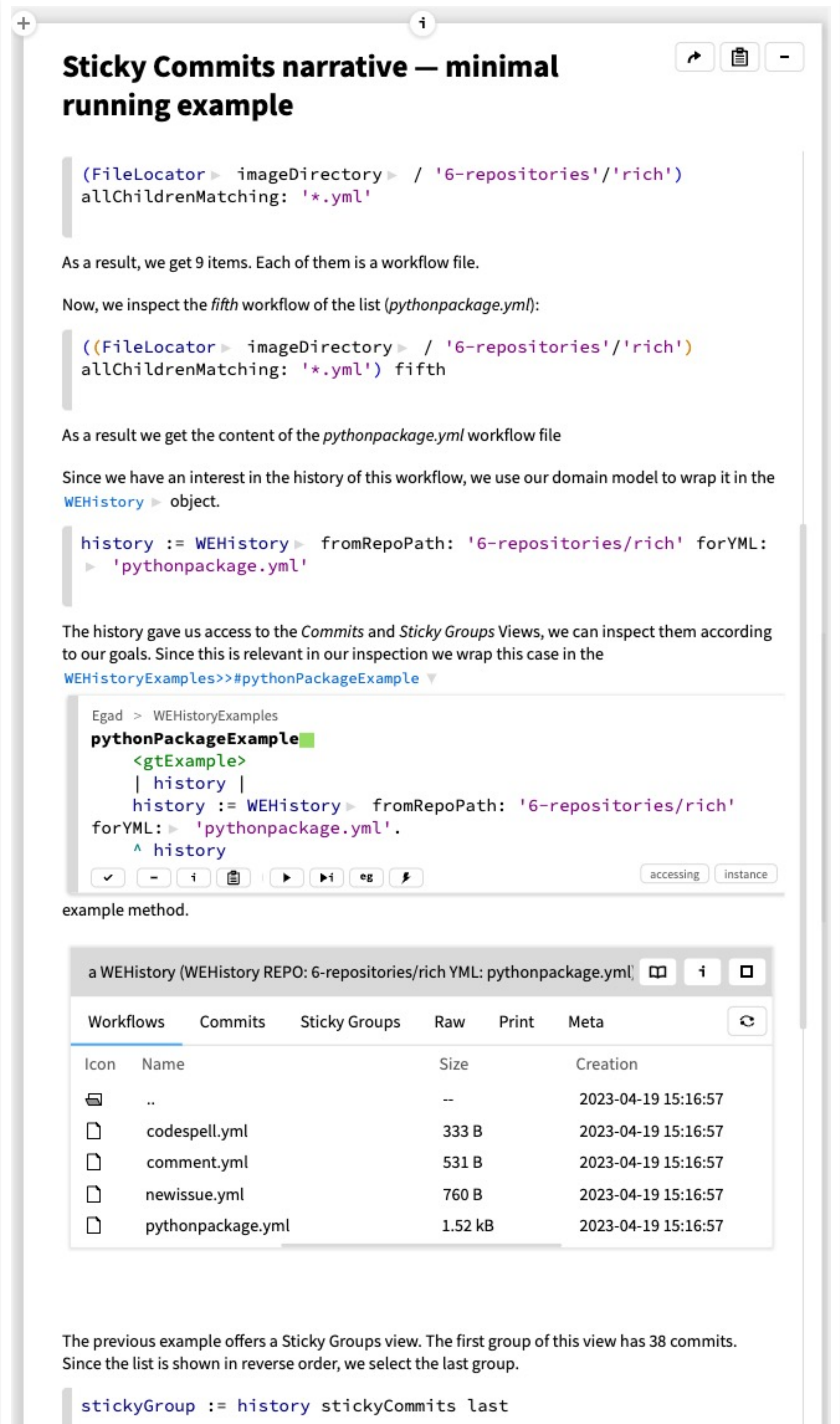
Problem of this approach

1. No context
2. No domain model

We cannot fully explore the Domain!

EGAD: Explorable GitHub Action Domain Model

glamoroustoolkit



Sticky Commits narrative — minimal running example

```
(FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` ▸ object.

```
history := WEHistory ▸ fromRepoPath: '6-repositories/rich' forYML:
▸ 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample` ▾

```
Egad > WEHistoryExamples
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory ▸ fromRepoPath: '6-repositories/rich'
forYML: ▸ 'pythonpackage.yml'.
    ^ history
```

example method.

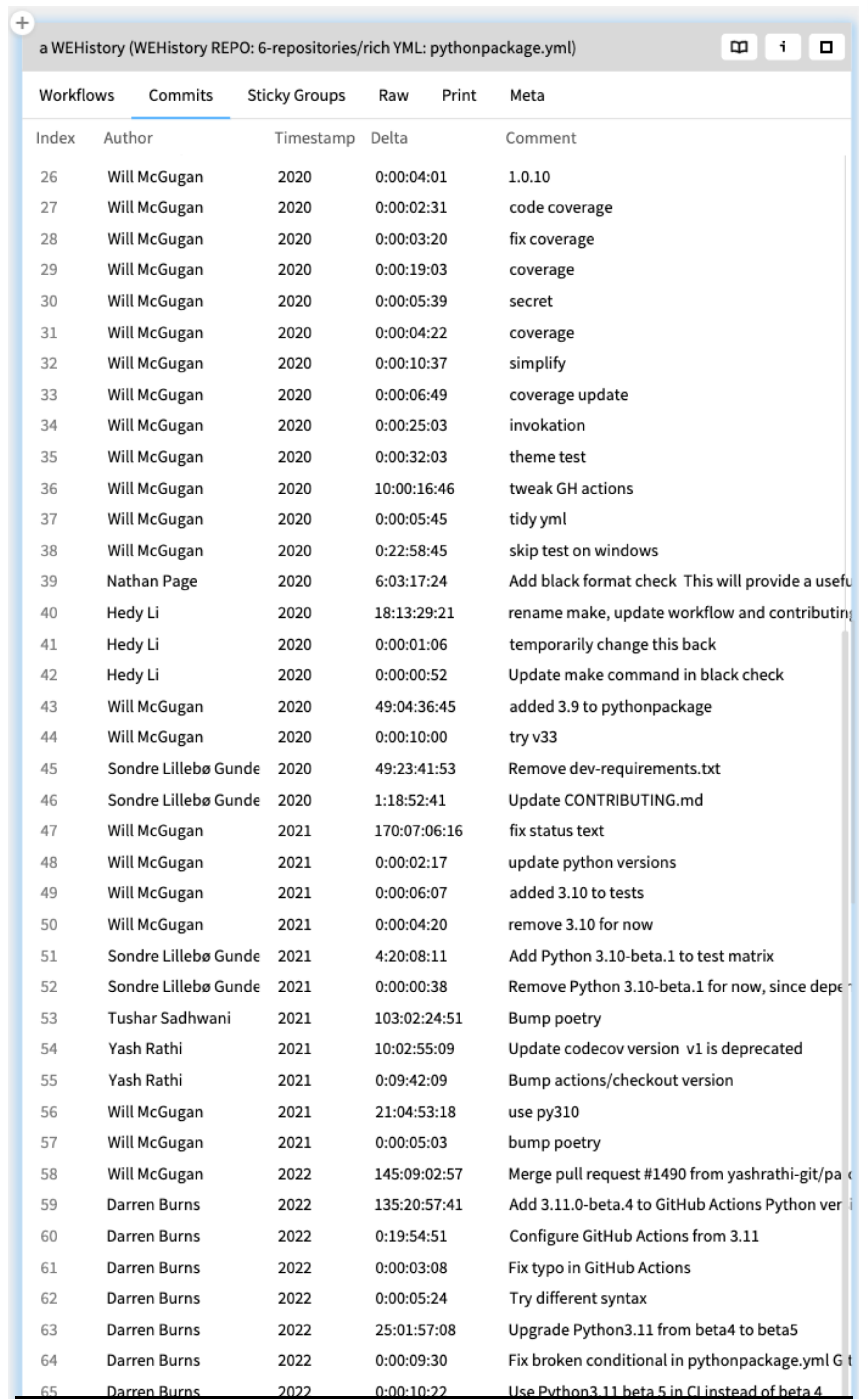
a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
🗀	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

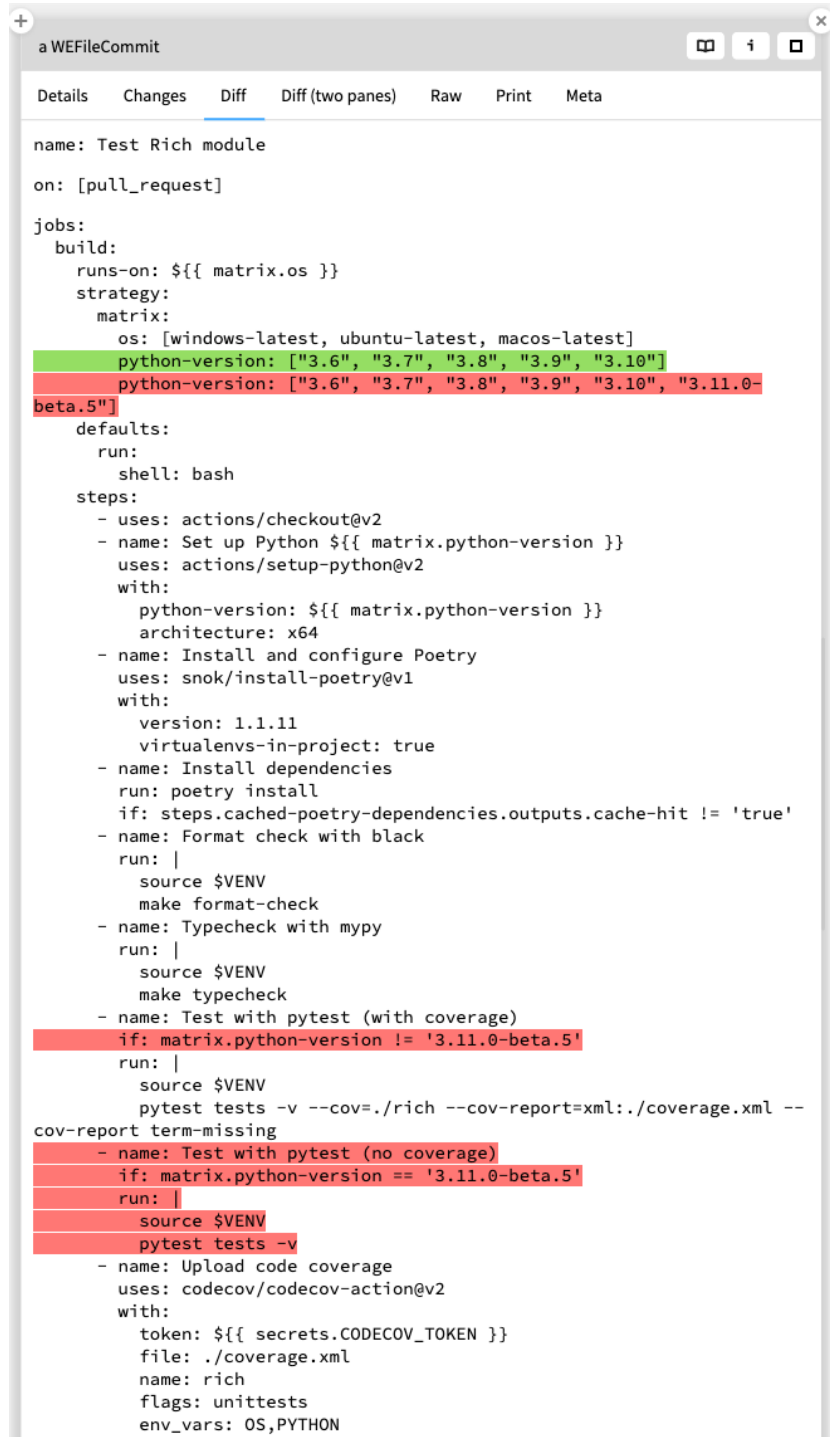
Narratives



a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Index	Author	Timestamp	Delta	Comment
26	Will McGugan	2020	0:00:04:01	1.0.10
27	Will McGugan	2020	0:00:02:31	code coverage
28	Will McGugan	2020	0:00:03:20	fix coverage
29	Will McGugan	2020	0:00:19:03	coverage
30	Will McGugan	2020	0:00:05:39	secret
31	Will McGugan	2020	0:00:04:22	coverage
32	Will McGugan	2020	0:00:10:37	simplify
33	Will McGugan	2020	0:00:06:49	coverage update
34	Will McGugan	2020	0:00:25:03	invokation
35	Will McGugan	2020	0:00:32:03	theme test
36	Will McGugan	2020	10:00:16:46	tweak GH actions
37	Will McGugan	2020	0:00:05:45	tidy yml
38	Will McGugan	2020	0:22:58:45	skip test on windows
39	Nathan Page	2020	6:03:17:24	Add black format check This will provide a usefu
40	Hedy Li	2020	18:13:29:21	rename make, update workflow and contributin
41	Hedy Li	2020	0:00:01:06	temporarily change this back
42	Hedy Li	2020	0:00:00:52	Update make command in black check
43	Will McGugan	2020	49:04:36:45	added 3.9 to pythonpackage
44	Will McGugan	2020	0:00:10:00	try v33
45	Sondre Lillebø Gunde	2020	49:23:41:53	Remove dev-requirements.txt
46	Sondre Lillebø Gunde	2020	1:18:52:41	Update CONTRIBUTING.md
47	Will McGugan	2021	170:07:06:16	fix status text
48	Will McGugan	2021	0:00:02:17	update python versions
49	Will McGugan	2021	0:00:06:07	added 3.10 to tests
50	Will McGugan	2021	0:00:04:20	remove 3.10 for now
51	Sondre Lillebø Gunde	2021	4:20:08:11	Add Python 3.10-beta.1 to test matrix
52	Sondre Lillebø Gunde	2021	0:00:00:38	Remove Python 3.10-beta.1 for now, since depe
53	Tushar Sadhwani	2021	103:02:24:51	Bump poetry
54	Yash Rathi	2021	10:02:55:09	Update codecov version v1 is deprecated
55	Yash Rathi	2021	0:09:42:09	Bump actions/checkout version
56	Will McGugan	2021	21:04:53:18	use py310
57	Will McGugan	2021	0:00:05:03	bump poetry
58	Will McGugan	2022	145:09:02:57	Merge pull request #1490 from yashrathi-git/pa
59	Darren Burns	2022	135:20:57:41	Add 3.11.0-beta.4 to GitHub Actions Python ver
60	Darren Burns	2022	0:19:54:51	Configure GitHub Actions from 3.11
61	Darren Burns	2022	0:00:03:08	Fix typo in GitHub Actions
62	Darren Burns	2022	0:00:05:24	Try different syntax
63	Darren Burns	2022	25:01:57:08	Upgrade Python3.11 from beta4 to beta5
64	Darren Burns	2022	0:00:09:30	Fix broken conditional in pythonpackage.yml G
65	Darren Burns	2022	0:00:10:22	Use Python3.11 beta 5 in CI instead of beta 4

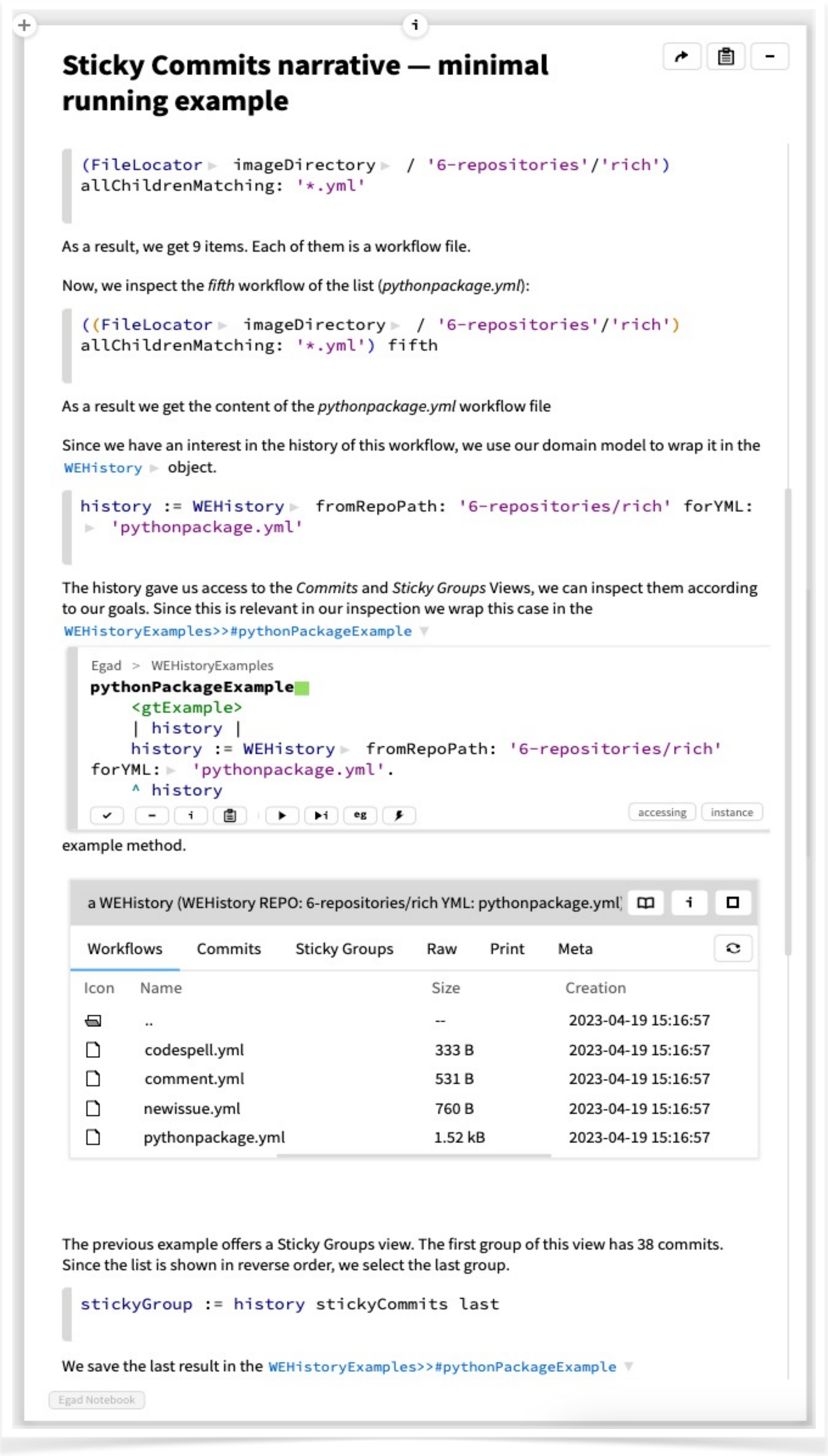
GA Domain Model



a WEFileCommit

```
name: Test Rich module

on: [pull_request]

jobs:
  build:
    runs-on: ${{ matrix.os }}
    strategy:
      matrix:
        os: [windows-latest, ubuntu-latest, macos-latest]
        python-version: ["3.6", "3.7", "3.8", "3.9", "3.10"]
        python-version: ["3.6", "3.7", "3.8", "3.9", "3.10", "3.11.0-
beta.5"]
    defaults:
      run:
        shell: bash
    steps:
      - uses: actions/checkout@v2
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${{ matrix.python-version }}
          architecture: x64
      - name: Install and configure Poetry
        uses: snok/install-poetry@v1
        with:
          version: 1.1.11
          virtualenvs-in-project: true
      - name: Install dependencies
        run: poetry install
        if: steps.cached-poetry-dependencies.outputs.cache-hit != 'true'
      - name: Format check with black
        run: |
          source $VENV
          make format-check
      - name: Typecheck with mypy
        run: |
          source $VENV
          make typecheck
      - name: Test with pytest (with coverage)
        if: matrix.python-version != '3.11.0-beta.5'
        run: |
          source $VENV
          pytest tests -v --cov=./rich --cov-report=xml:./coverage.xml --
cov-report term-missing
      - name: Test with pytest (no coverage)
        if: matrix.python-version == '3.11.0-beta.5'
        run: |
          source $VENV
          pytest tests -v
      - name: Upload code coverage
        uses: codecov/codecov-action@v2
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
          file: ./coverage.xml
          name: rich
          flags: unittests
          env_vars: OS,PYTHON
```

Custom Views



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample`

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
	..	--	2023-04-19 15:16:57
	codespell.yml	333 B	2023-04-19 15:16:57
	comment.yml	531 B	2023-04-19 15:16:57
	newissue.yml	760 B	2023-04-19 15:16:57
	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

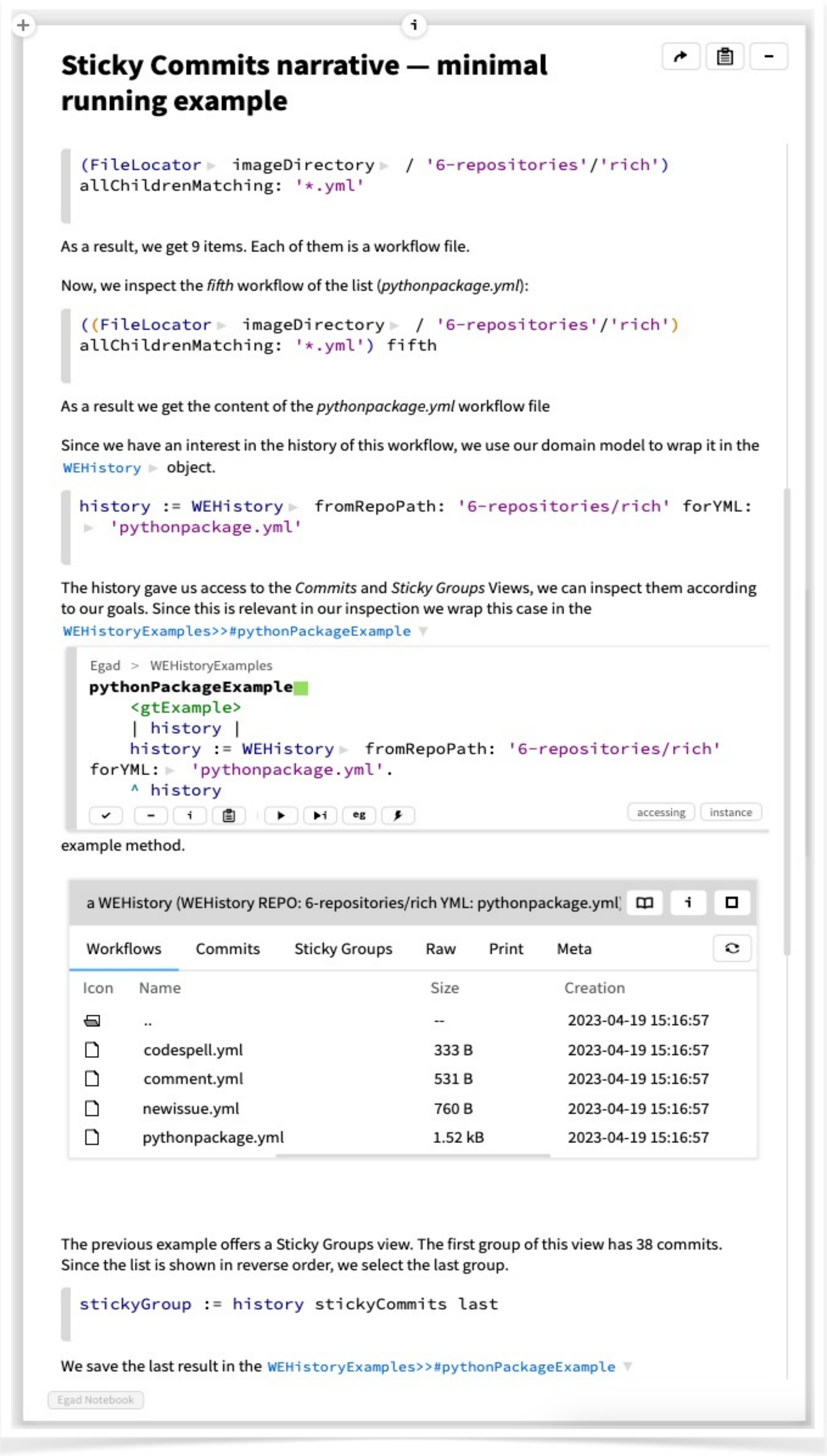
```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample`

Lesson 1:

Compose a story:

- *Documenting our tasks and progress, and*
- *Linking documentation and source code*



Sticky Commits narrative — minimal running example

```
(FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` ▸ object.

```
history := WEHistory ▸ fromRepoPath: '6-repositories/rich' forYML:
▸ 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample` ▾

```
Egad > WEHistoryExamples
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory ▸ fromRepoPath: '6-repositories/rich'
forYML: ▸ 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

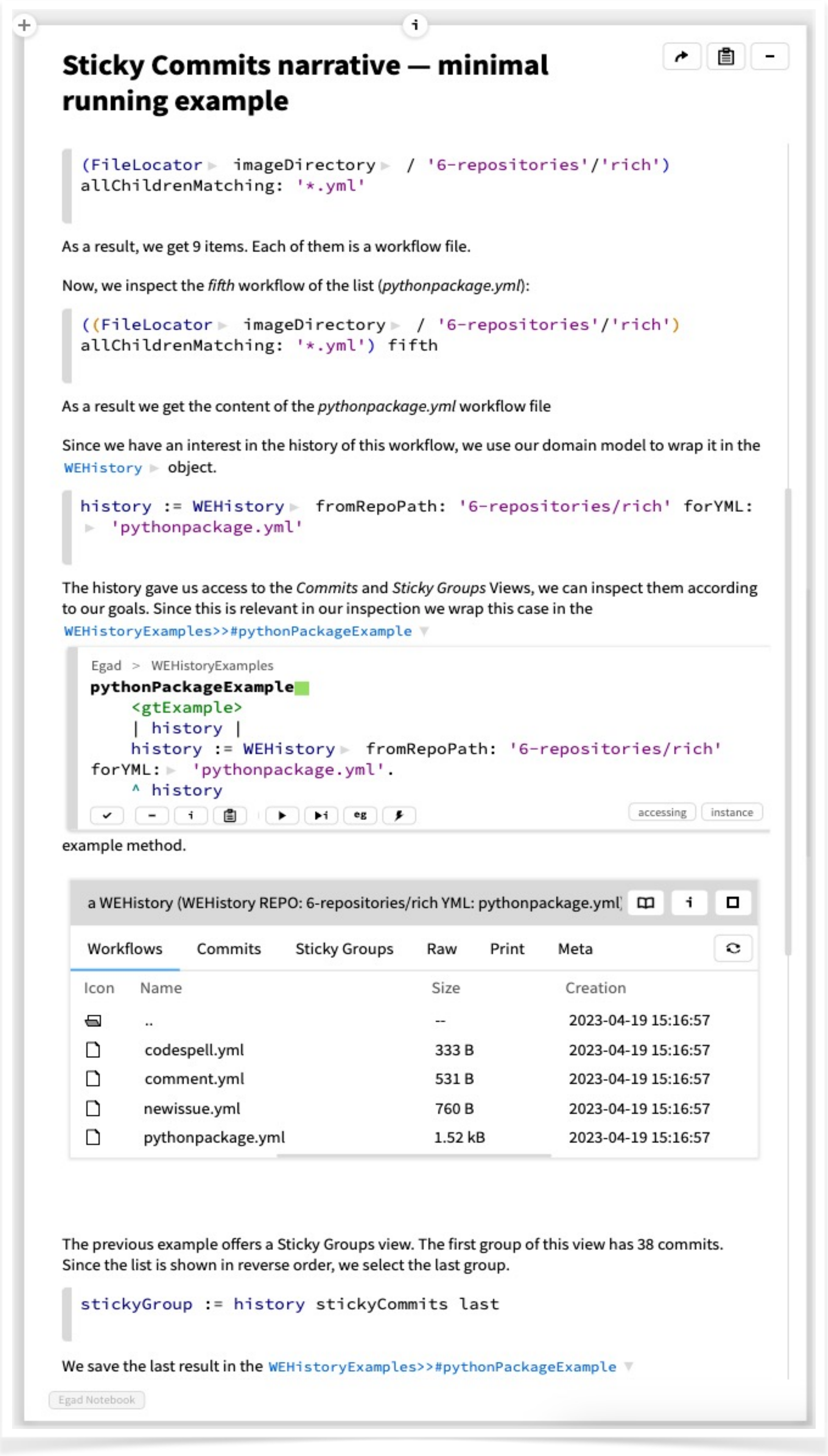
Icon	Name	Size	Creation
🗀	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample` ▾

Inspect the “rich” repository



Sticky Commits narrative — minimal running example

```
(FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` ▸ object.

```
history := WEHistory ▸ fromRepoPath: '6-repositories/rich' forYML:
▸ 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample` ▼

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory ▸ fromRepoPath: '6-repositories/rich'
forYML: ▸ 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
🗁	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

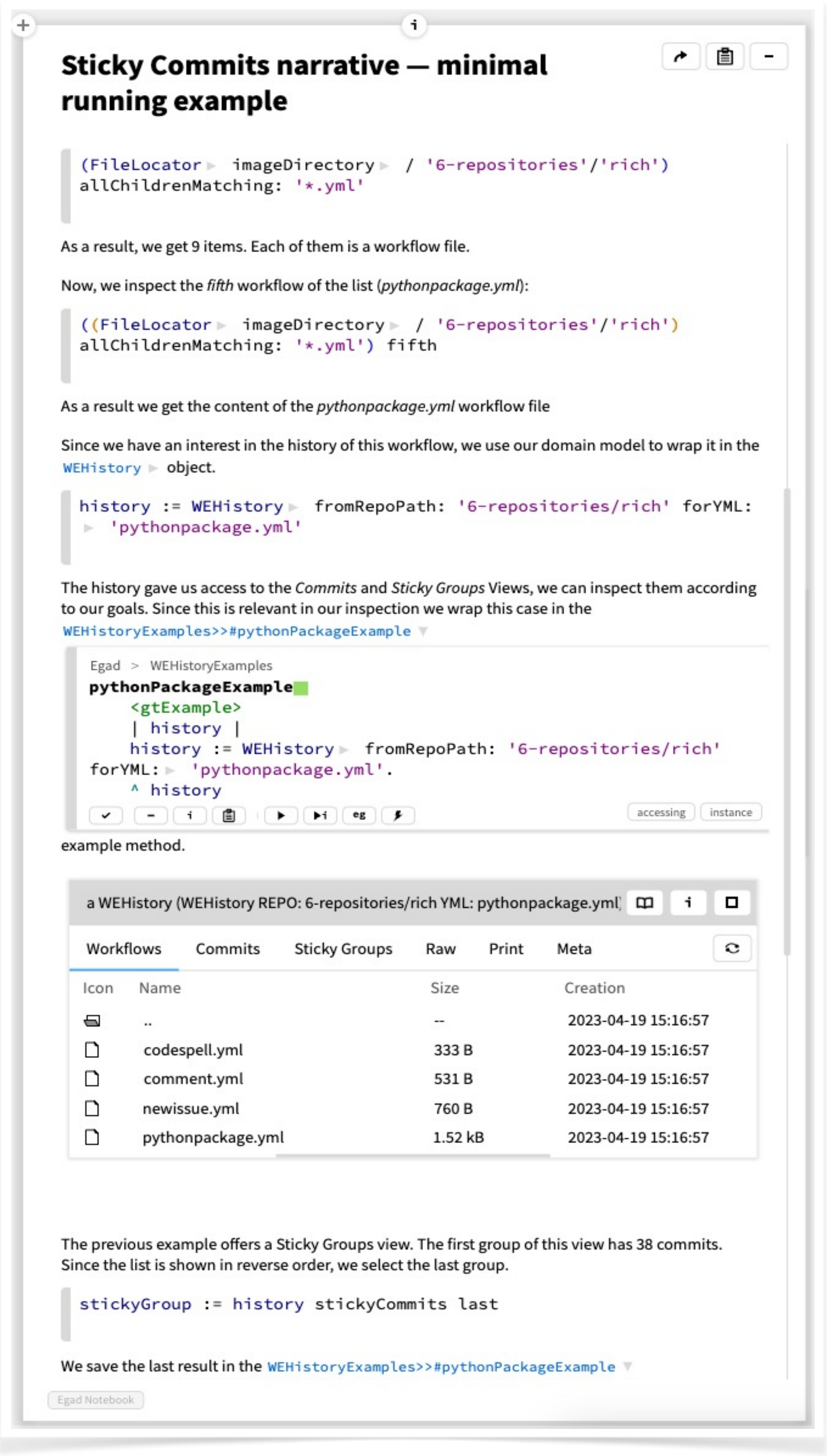
The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample` ▼

← Inspect the “rich” repository

← Identify all the YAML files



Sticky Commits narrative — minimal running example

```
(FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` ▸ object.

```
history := WEHistory ▸ fromRepoPath: '6-repositories/rich' forYML:
▸ 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample` ▾

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory ▸ fromRepoPath: '6-repositories/rich'
forYML: ▸ 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
🗀	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

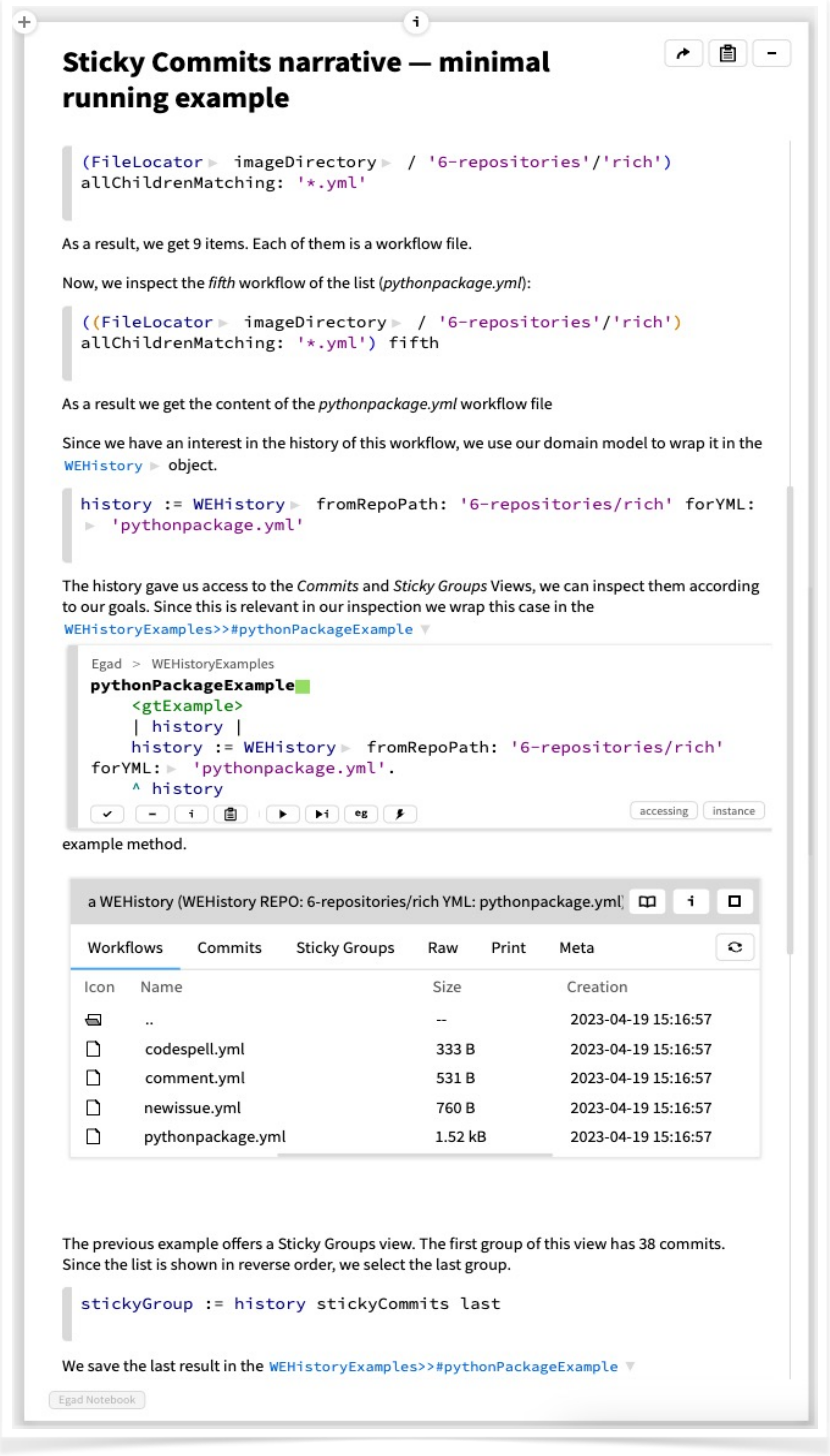
```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample` ▾

Inspect the “rich” repository

Identify all the YAML files

Wrap the YML file using the domain model (*pythonpackage.yml*)



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample` example method.

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
	..	--	2023-04-19 15:16:57
	codespell.yml	333 B	2023-04-19 15:16:57
	comment.yml	531 B	2023-04-19 15:16:57
	newissue.yml	760 B	2023-04-19 15:16:57
	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

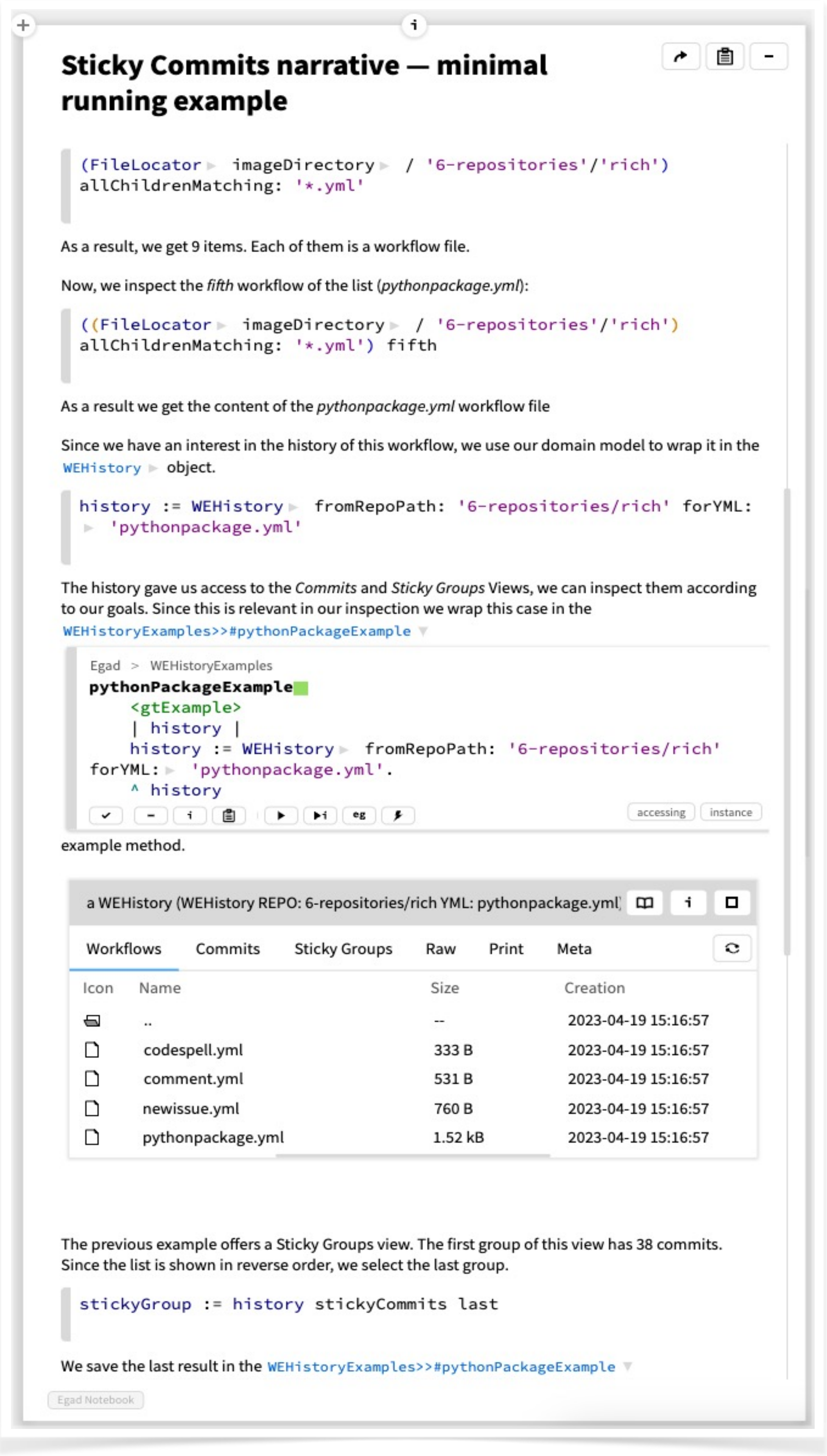
We save the last result in the `WEHistoryExamples>>#pythonPackageExample`

Inspect the "rich" repository

Identify all the YAML files

Wrap the YML file using the domain model (*pythonpackage.yml*)

Turn the domain model into an inspectable Example



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample`

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
🗀	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample`

Inspect the “rich” repository

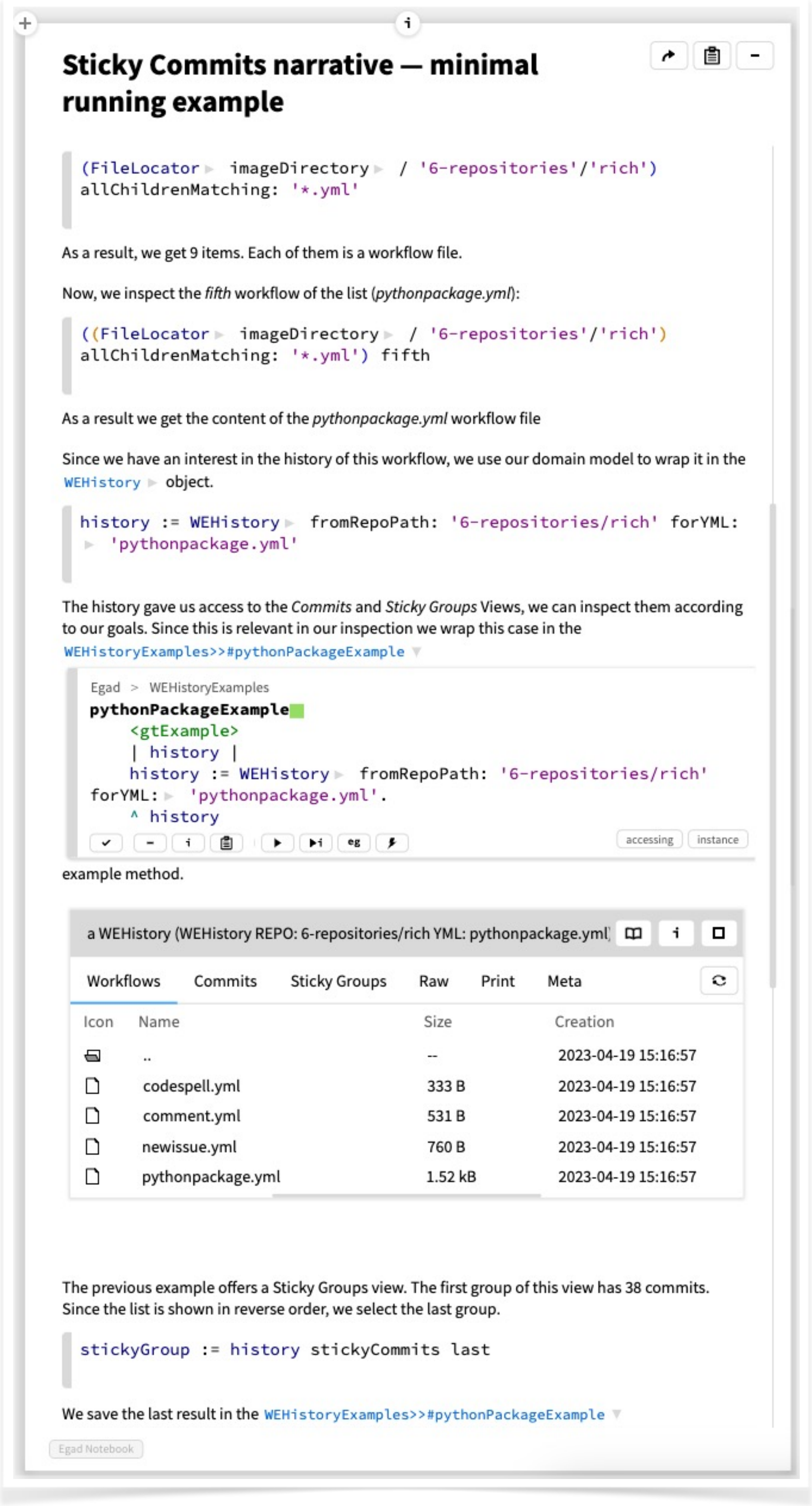
Identify all the YAML files

Wrap the YML file using the domain model (*pythonpackage.yml*)

Turn the domain model into an inspectable Example

Include the result of the example inspection

Navigating custom views



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the WEHistory object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the WEHistoryExamples>>#pythonPackageExample

```
Egad > WEHistoryExamples
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
	..	--	2023-04-19 15:16:57
	codespell.yml	333 B	2023-04-19 15:16:57
	comment.yml	531 B	2023-04-19 15:16:57
	newissue.yml	760 B	2023-04-19 15:16:57
	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

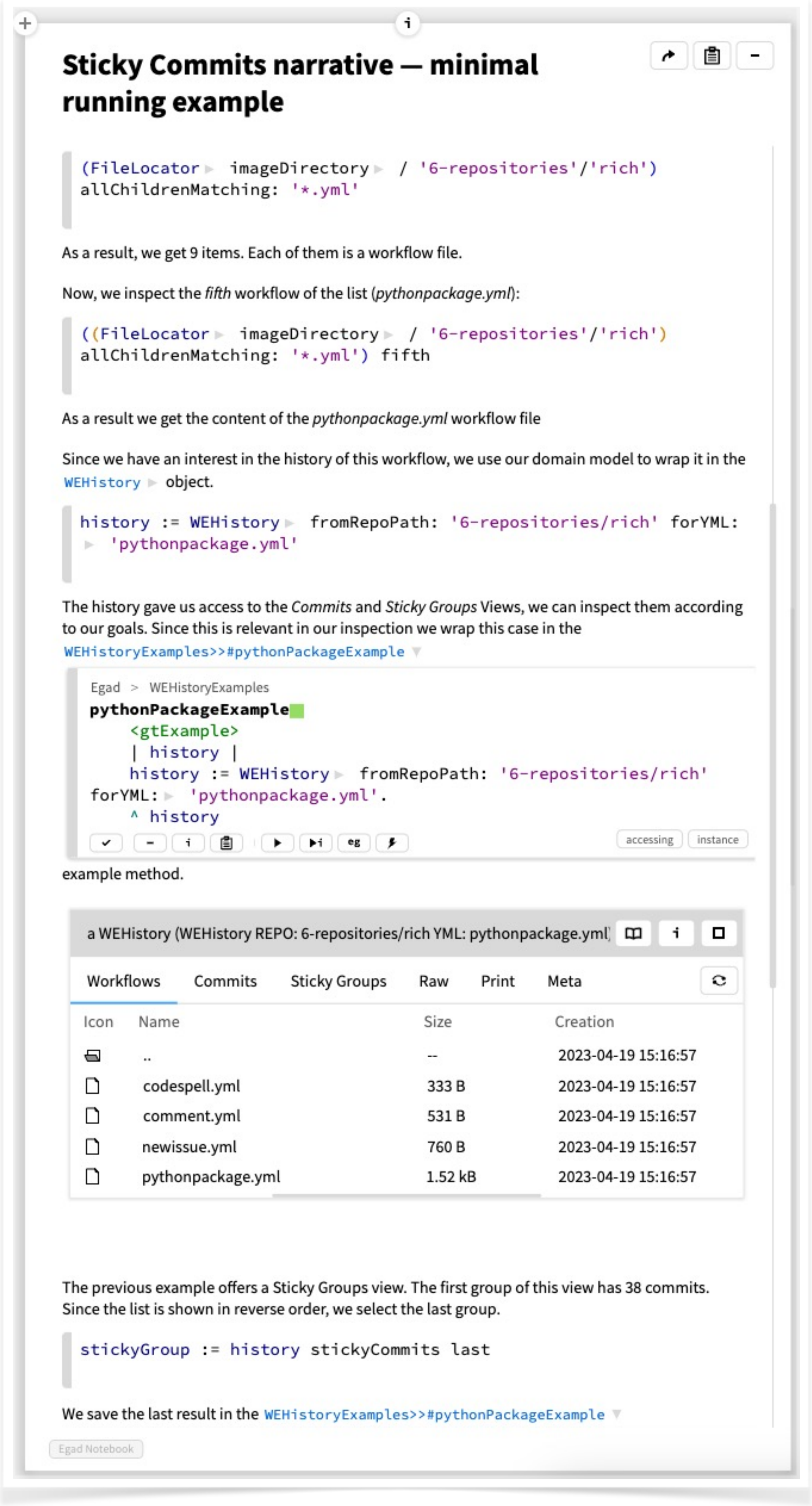
```
stickyGroup := history stickyCommits last
```

We save the last result in the WEHistoryExamples>>#pythonPackageExample

Lesson 2:

- ***Conduct research*** *by navigating custom views*

Navigating custom views



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the WEHistory object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the WEHistoryExamples>>#pythonPackageExample

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

accessing instance

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Workflows Commits Sticky Groups Raw Print Meta

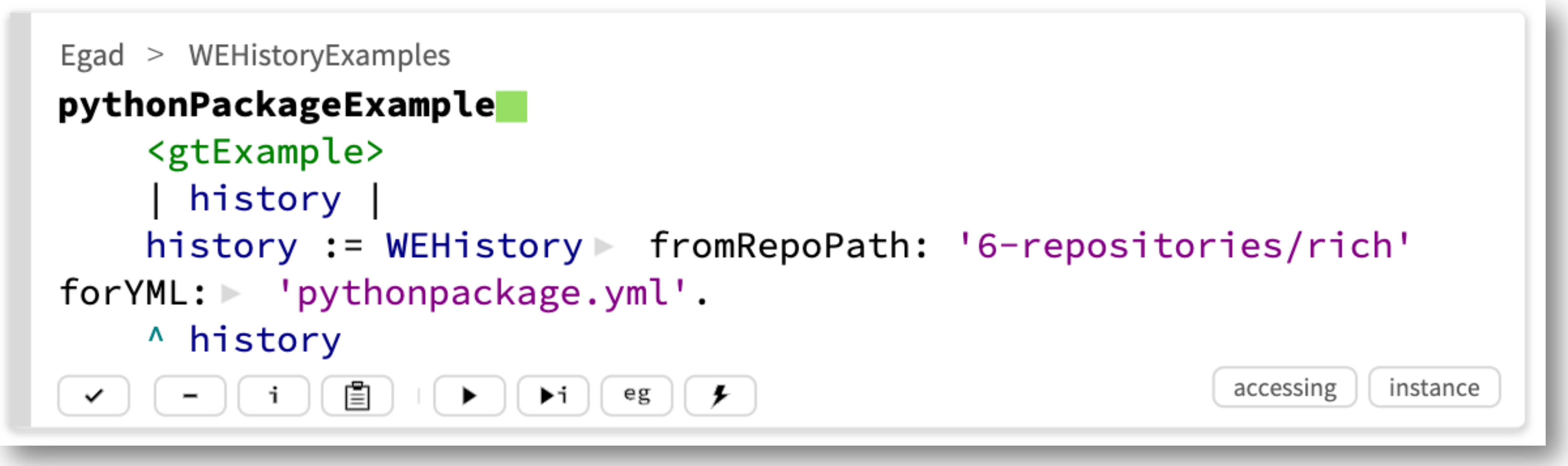
Icon	Name	Size	Creation
🗀	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

We save the last result in the WEHistoryExamples>>#pythonPackageExample

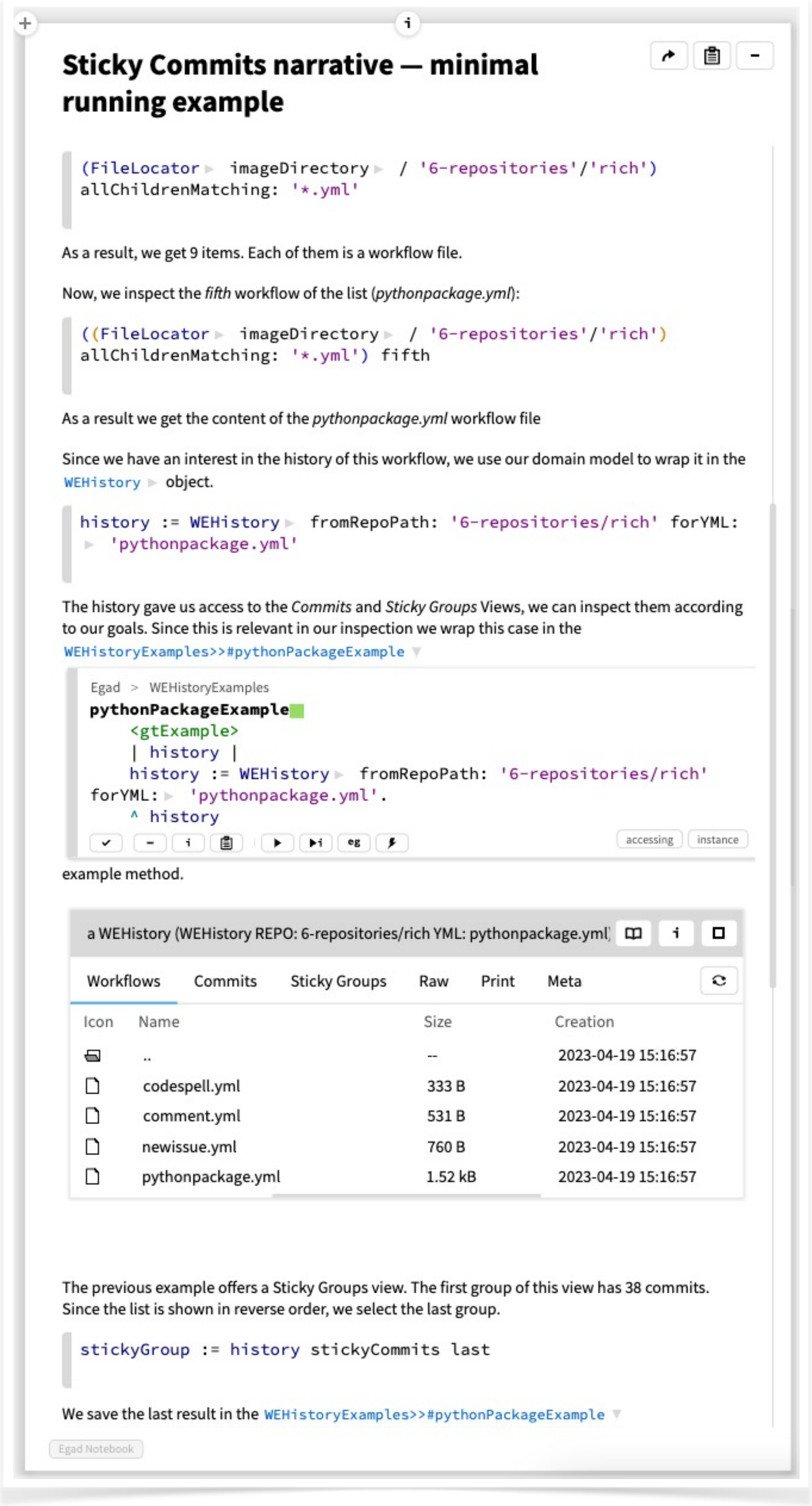
Egad Notebook



Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

Navigating custom views



Sticky Commits narrative — minimal running example

```
(FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` ▸ object.

```
history := WEHistory ▸ fromRepoPath: '6-repositories/rich' forYML:
▸ 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample` ▾

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory ▸ fromRepoPath: '6-repositories/rich'
forYML: ▸ 'pythonpackage.yml'.
    ^ history
```

example method.

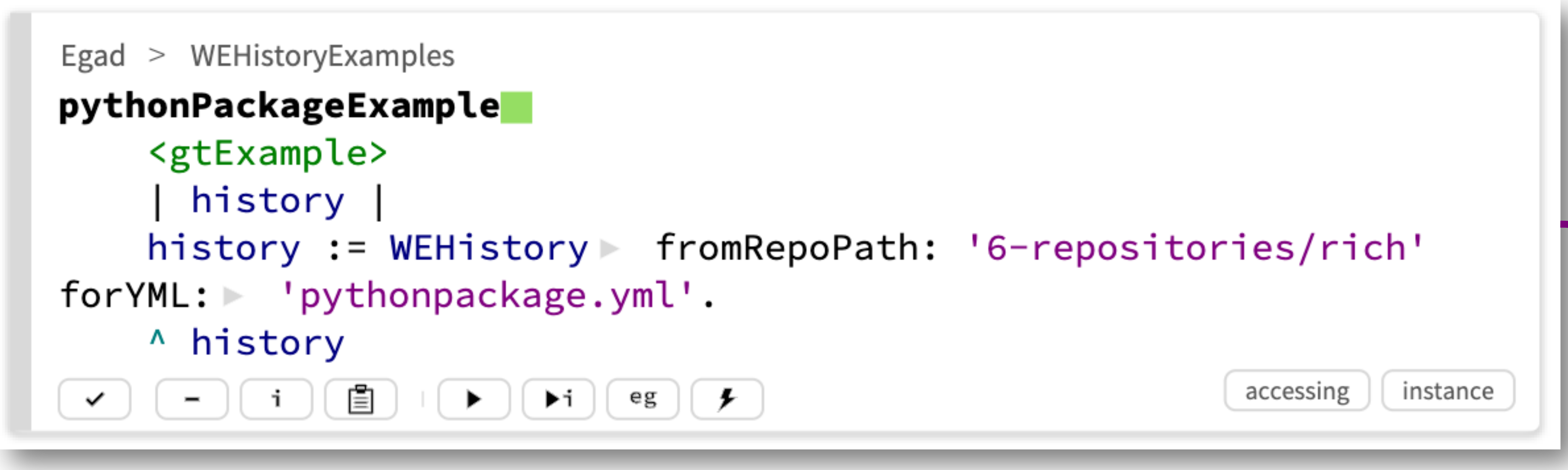
a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
🗀	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

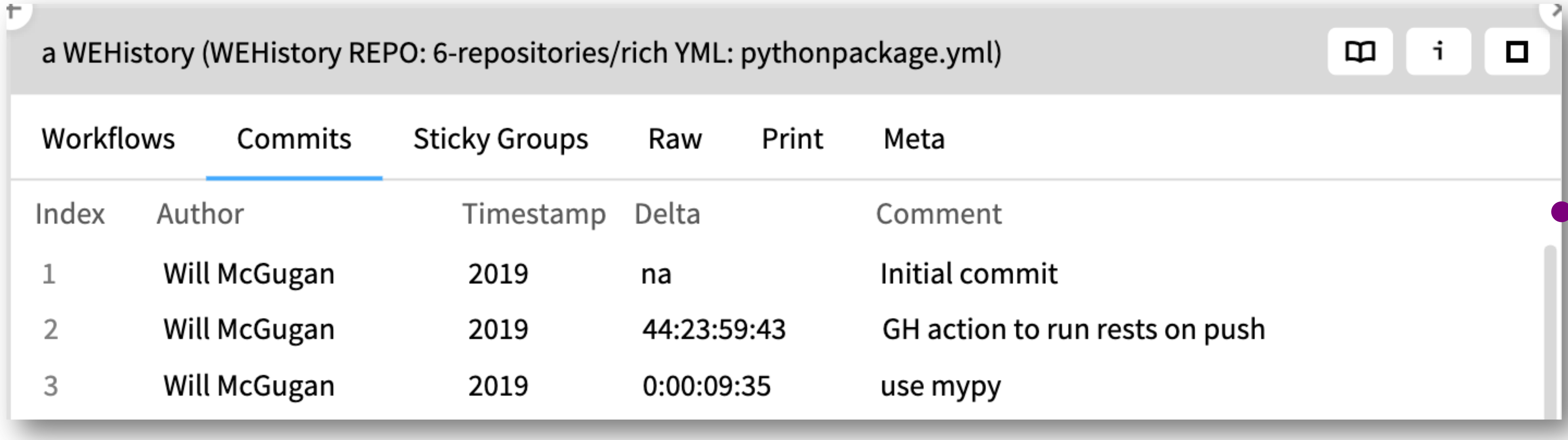
```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample` ▾



Egad > WEHistoryExamples

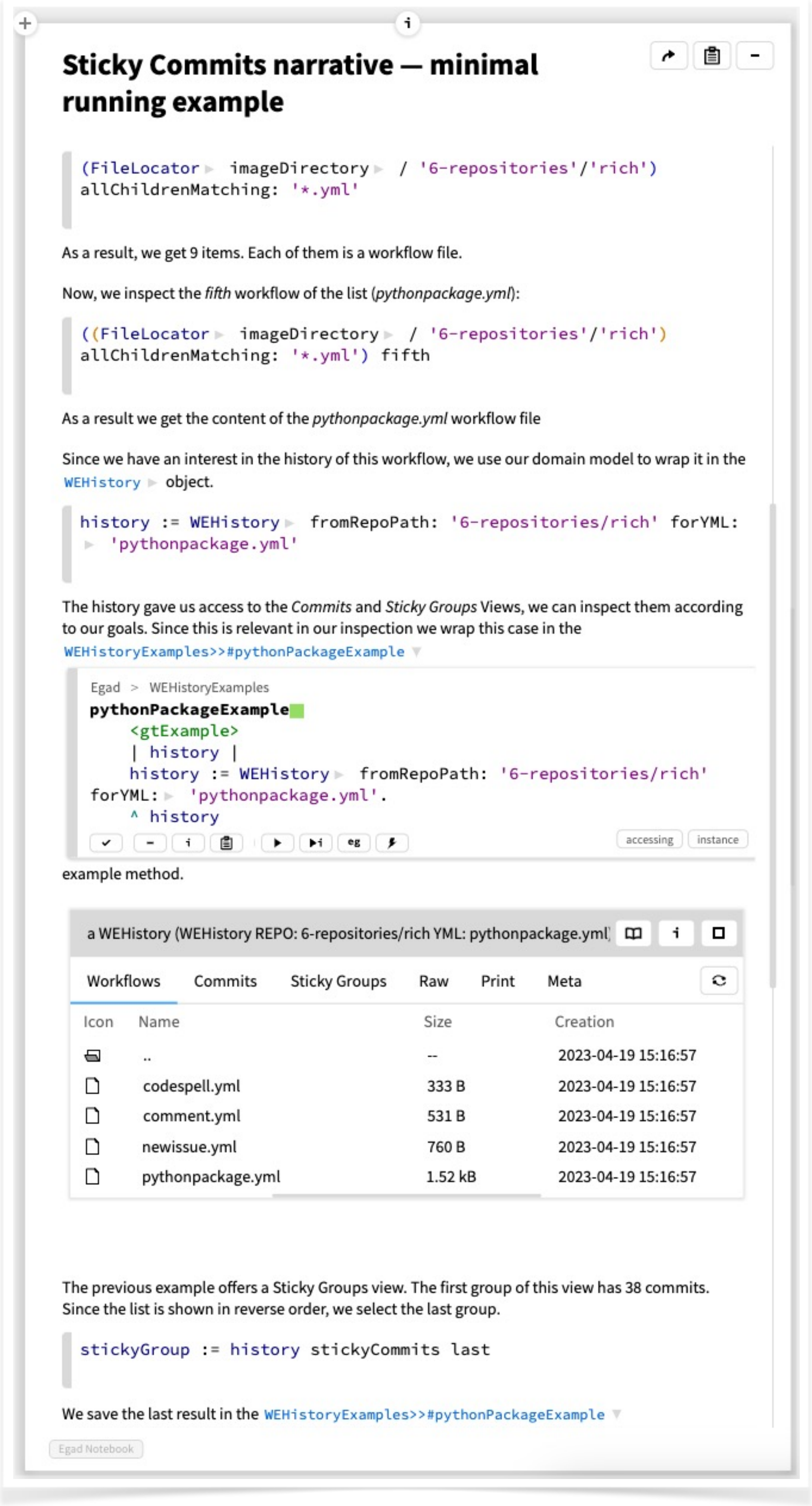
```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory ▸ fromRepoPath: '6-repositories/rich'
forYML: ▸ 'pythonpackage.yml'.
    ^ history
```



a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Index	Author	Timestamp	Delta	Comment
1	Will McGugan	2019	na	Initial commit
2	Will McGugan	2019	44:23:59:43	GH action to run rests on push
3	Will McGugan	2019	0:00:09:35	use mypy

Navigating custom views



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample`

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

example method.

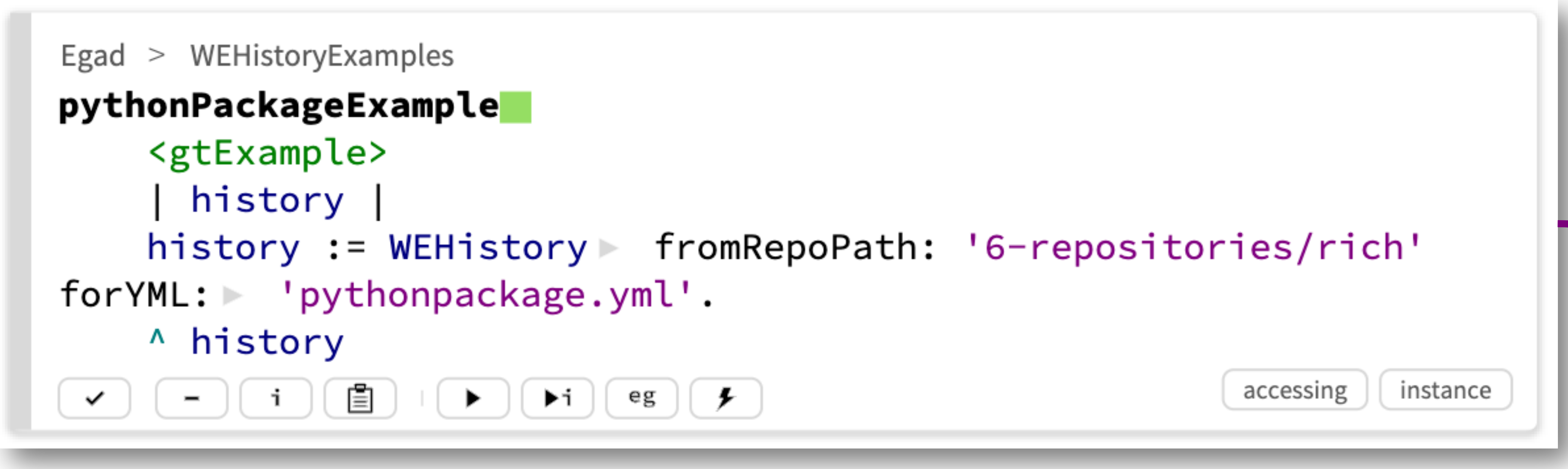
a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
	..	--	2023-04-19 15:16:57
	codespell.yml	333 B	2023-04-19 15:16:57
	comment.yml	531 B	2023-04-19 15:16:57
	newissue.yml	760 B	2023-04-19 15:16:57
	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

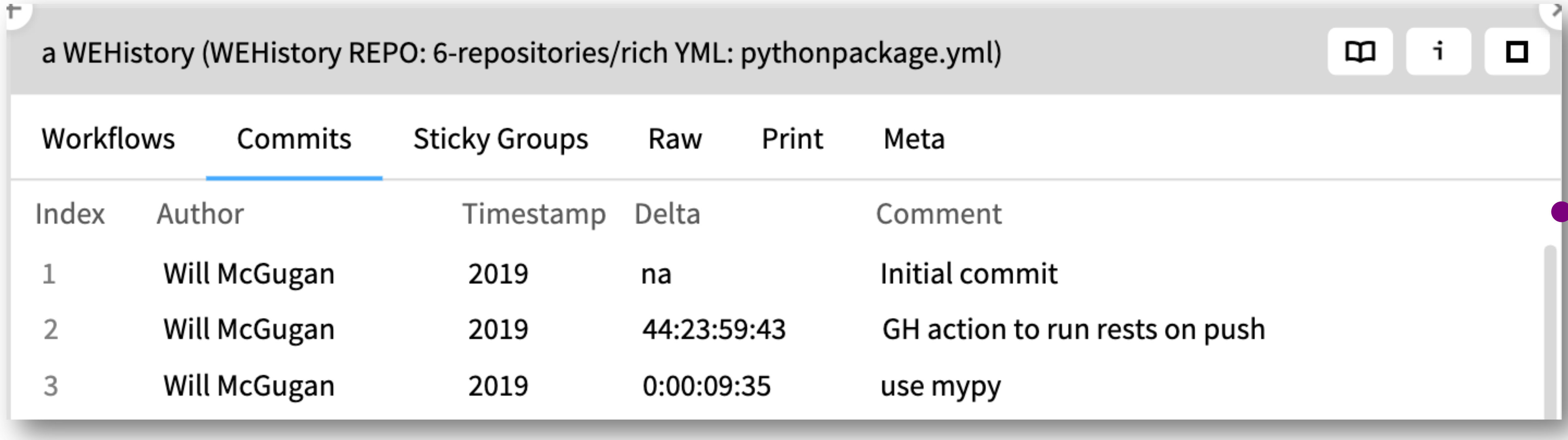
```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample`



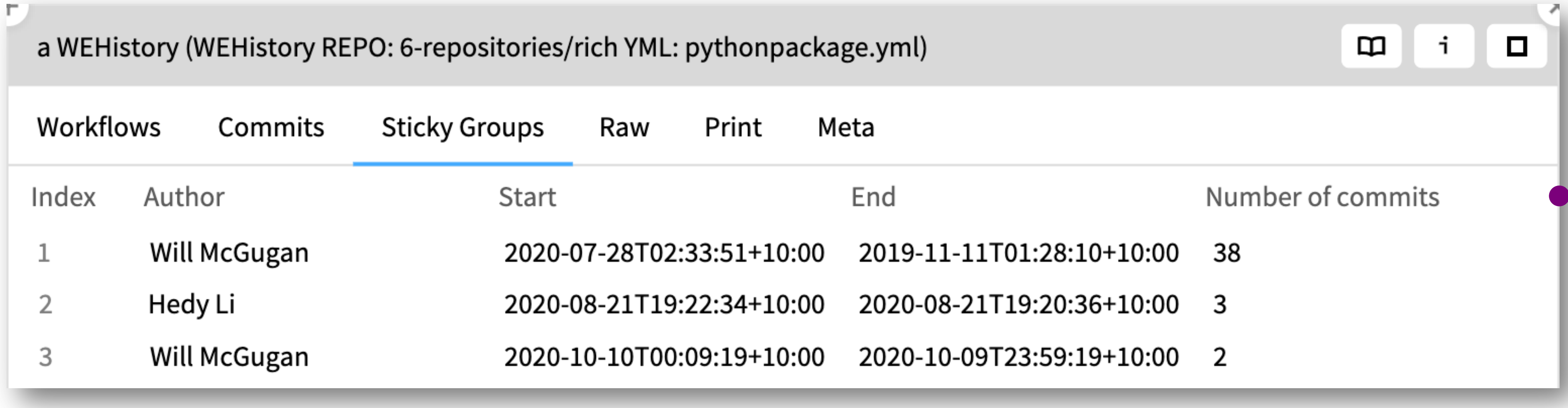
Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```



a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

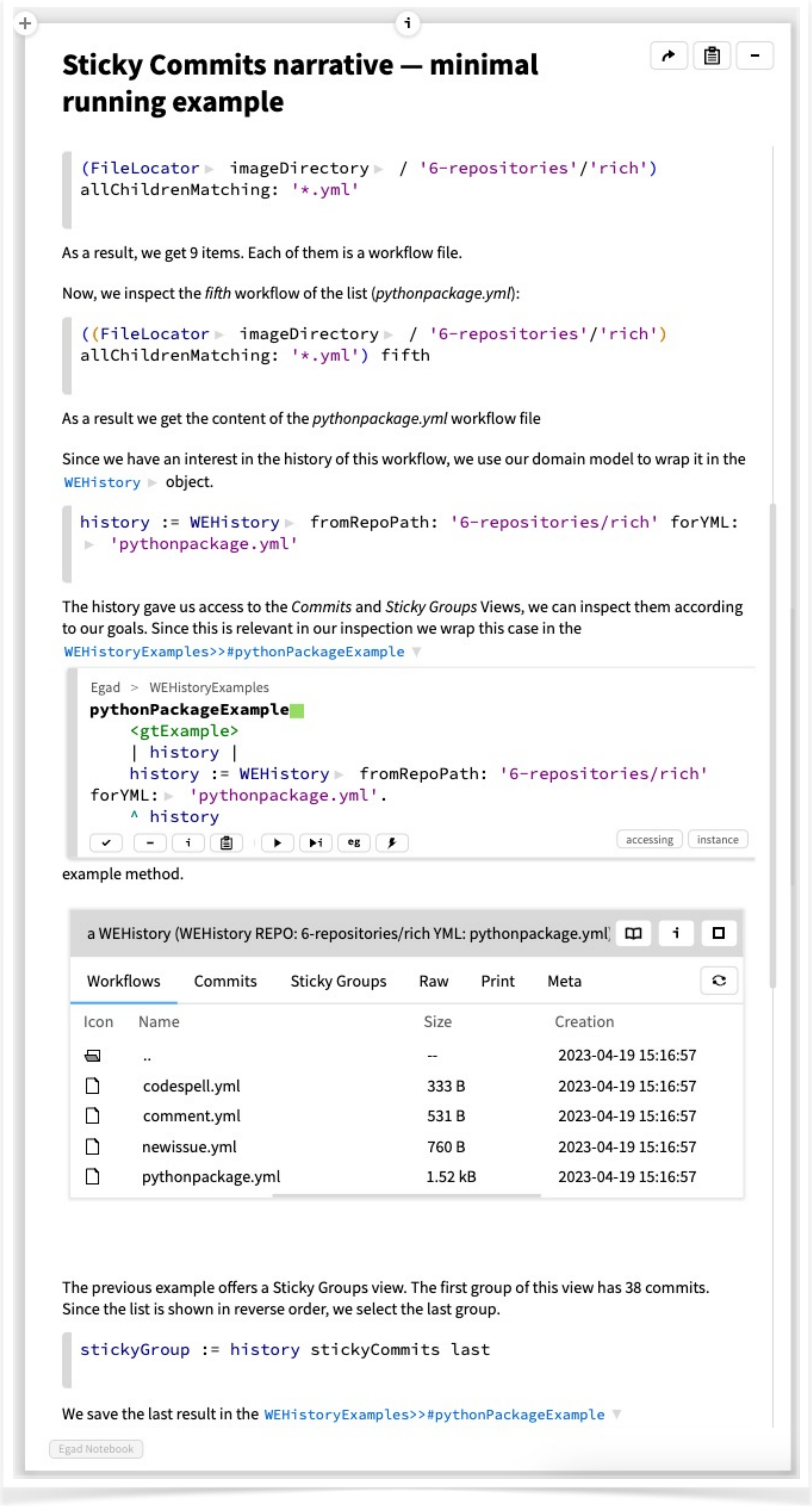
Index	Author	Timestamp	Delta	Comment
1	Will McGugan	2019	na	Initial commit
2	Will McGugan	2019	44:23:59:43	GH action to run rests on push
3	Will McGugan	2019	0:00:09:35	use mypy



a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Index	Author	Start	End	Number of commits
1	Will McGugan	2020-07-28T02:33:51+10:00	2019-11-11T01:28:10+10:00	38
2	Hedy Li	2020-08-21T19:22:34+10:00	2020-08-21T19:20:36+10:00	3
3	Will McGugan	2020-10-10T00:09:19+10:00	2020-10-09T23:59:19+10:00	2

Navigating custom views



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample`

```
Egad > WEHistoryExamples
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

example method.

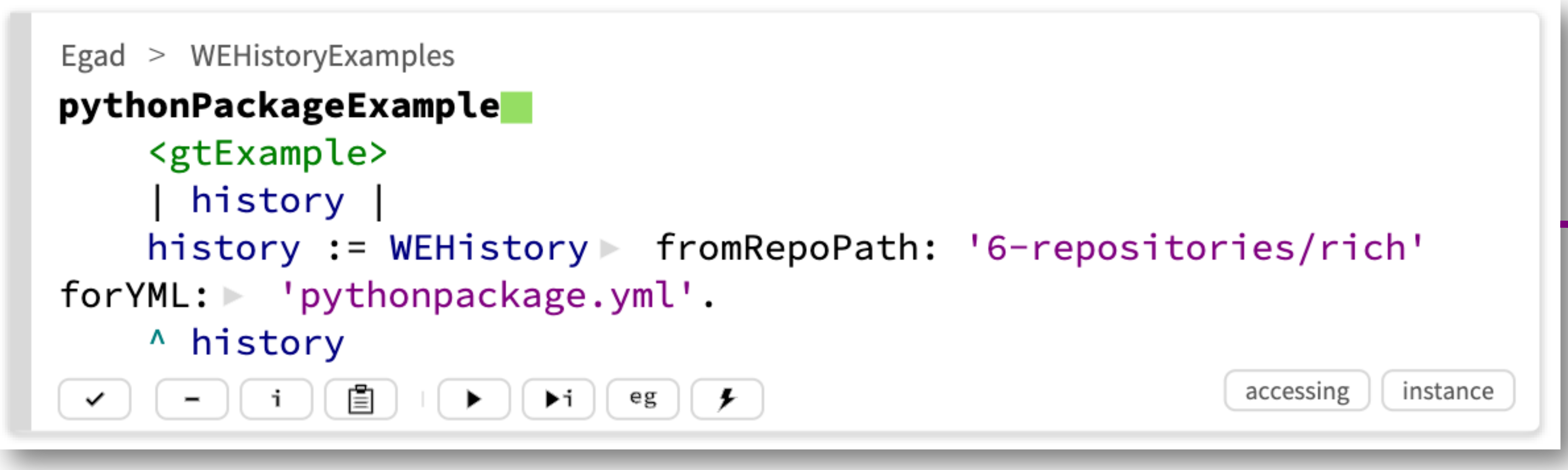
a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
	..	--	2023-04-19 15:16:57
	codespell.yml	333 B	2023-04-19 15:16:57
	comment.yml	531 B	2023-04-19 15:16:57
	newissue.yml	760 B	2023-04-19 15:16:57
	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

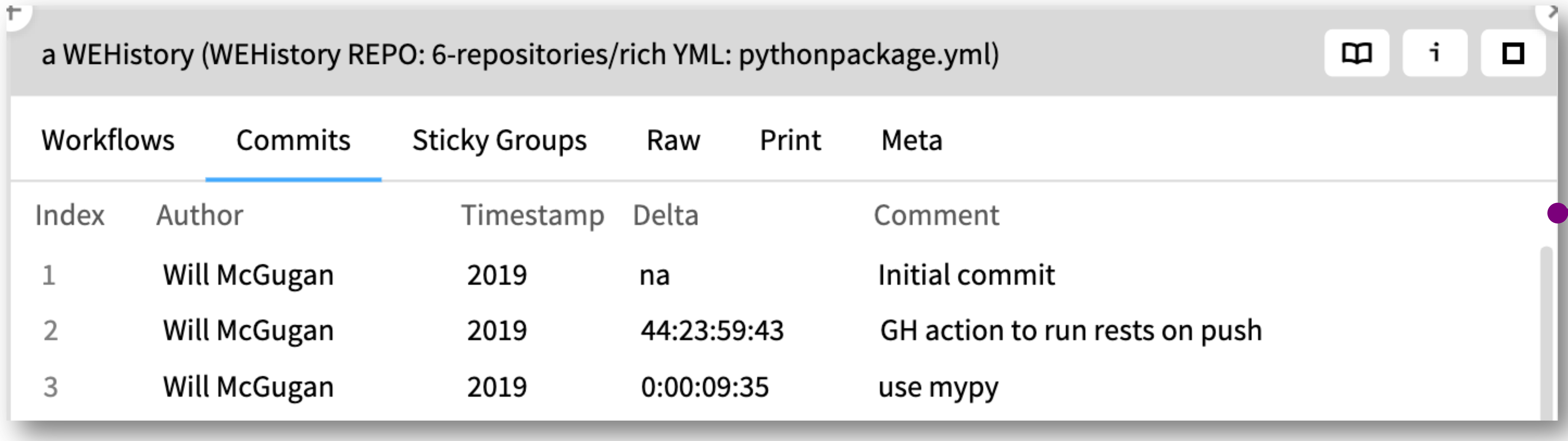
The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample`

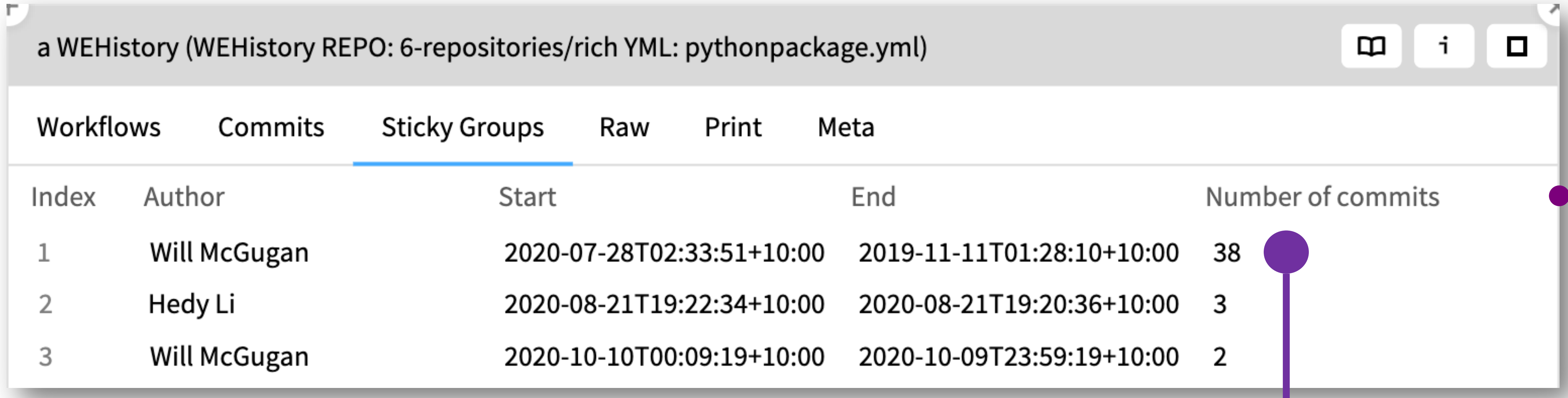


```
Egad > WEHistoryExamples
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```



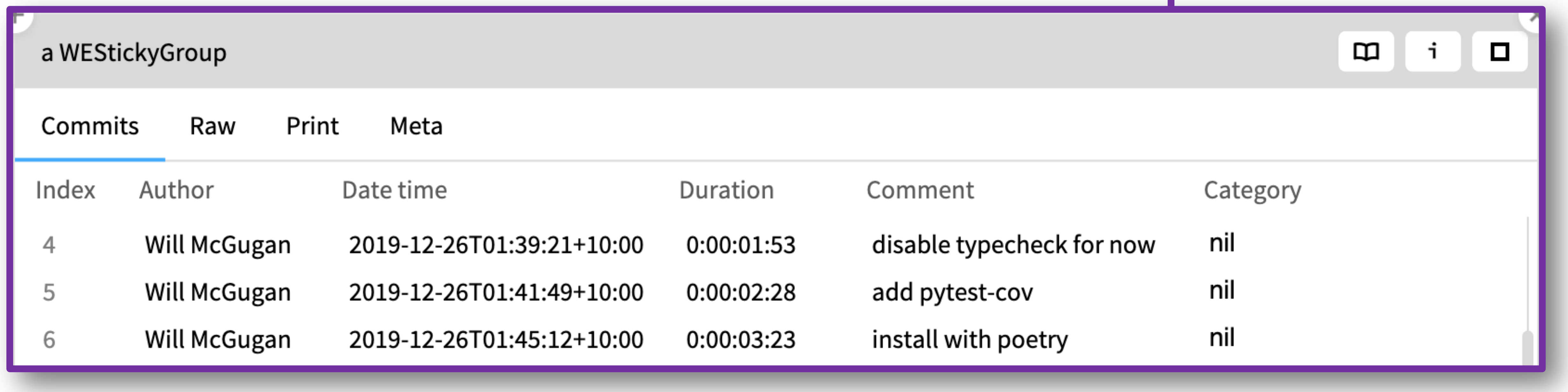
a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Index	Author	Timestamp	Delta	Comment
1	Will McGugan	2019	na	Initial commit
2	Will McGugan	2019	44:23:59:43	GH action to run rests on push
3	Will McGugan	2019	0:00:09:35	use mypy



a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

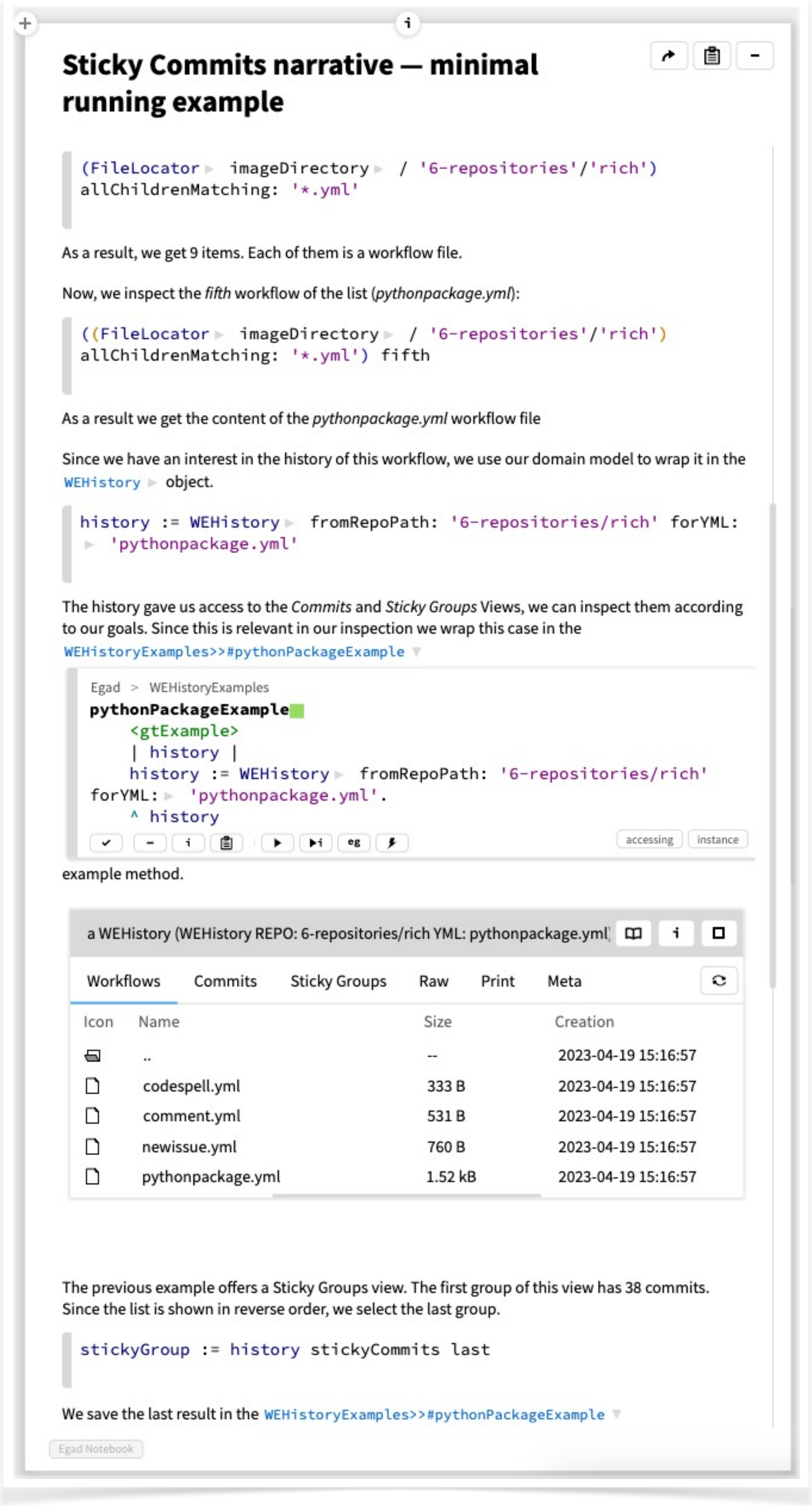
Index	Author	Start	End	Number of commits
1	Will McGugan	2020-07-28T02:33:51+10:00	2019-11-11T01:28:10+10:00	38
2	Hedy Li	2020-08-21T19:22:34+10:00	2020-08-21T19:20:36+10:00	3
3	Will McGugan	2020-10-10T00:09:19+10:00	2020-10-09T23:59:19+10:00	2



a WEStickyGroup

Index	Author	Date time	Duration	Comment	Category
4	Will McGugan	2019-12-26T01:39:21+10:00	0:00:01:53	disable typecheck for now	nil
5	Will McGugan	2019-12-26T01:41:49+10:00	0:00:02:28	add pytest-cov	nil
6	Will McGugan	2019-12-26T01:45:12+10:00	0:00:03:23	install with poetry	nil

Onboarding of researchers on GT



Sticky Commits narrative — minimal running example

```
(FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator ▸ imageDirectory ▸ / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory ▸` object.

```
history := WEHistory ▸ fromRepoPath: '6-repositories/rich' forYML:
▸ 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample ▼`

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory ▸ fromRepoPath: '6-repositories/rich'
forYML: ▸ 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
🗀	..	--	2023-04-19 15:16:57
🗋	codespell.yml	333 B	2023-04-19 15:16:57
🗋	comment.yml	531 B	2023-04-19 15:16:57
🗋	newissue.yml	760 B	2023-04-19 15:16:57
🗋	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

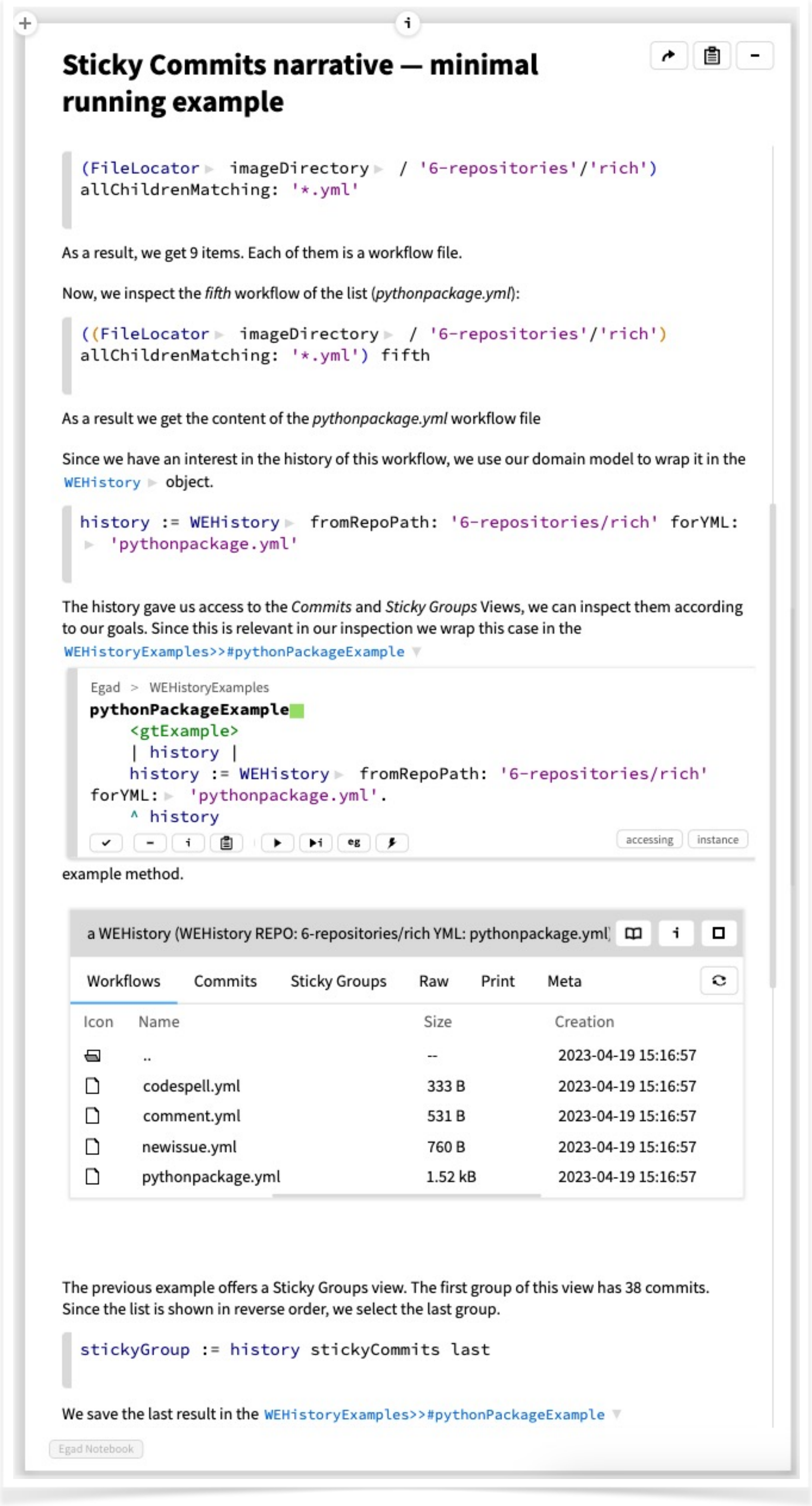
```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample ▼`

Lesson 3:

- ***Support** the onboarding!!*

Onboarding of researchers on GT



Sticky Commits narrative — minimal running example

```
(FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml'
```

As a result, we get 9 items. Each of them is a workflow file.

Now, we inspect the *fifth* workflow of the list (*pythonpackage.yml*):

```
((FileLocator imageDirectory / '6-repositories'/'rich')
allChildrenMatching: '*.yml') fifth
```

As a result we get the content of the *pythonpackage.yml* workflow file

Since we have an interest in the history of this workflow, we use our domain model to wrap it in the `WEHistory` object.

```
history := WEHistory fromRepoPath: '6-repositories/rich' forYML:
 'pythonpackage.yml'
```

The history gave us access to the *Commits* and *Sticky Groups* Views, we can inspect them according to our goals. Since this is relevant in our inspection we wrap this case in the `WEHistoryExamples>>#pythonPackageExample`

Egad > WEHistoryExamples

```
pythonPackageExample
    <gtExample>
    | history |
    history := WEHistory fromRepoPath: '6-repositories/rich'
forYML: 'pythonpackage.yml'.
    ^ history
```

example method.

a WEHistory (WEHistory REPO: 6-repositories/rich YML: pythonpackage.yml)

Icon	Name	Size	Creation
	..	--	2023-04-19 15:16:57
	codespell.yml	333 B	2023-04-19 15:16:57
	comment.yml	531 B	2023-04-19 15:16:57
	newissue.yml	760 B	2023-04-19 15:16:57
	pythonpackage.yml	1.52 kB	2023-04-19 15:16:57

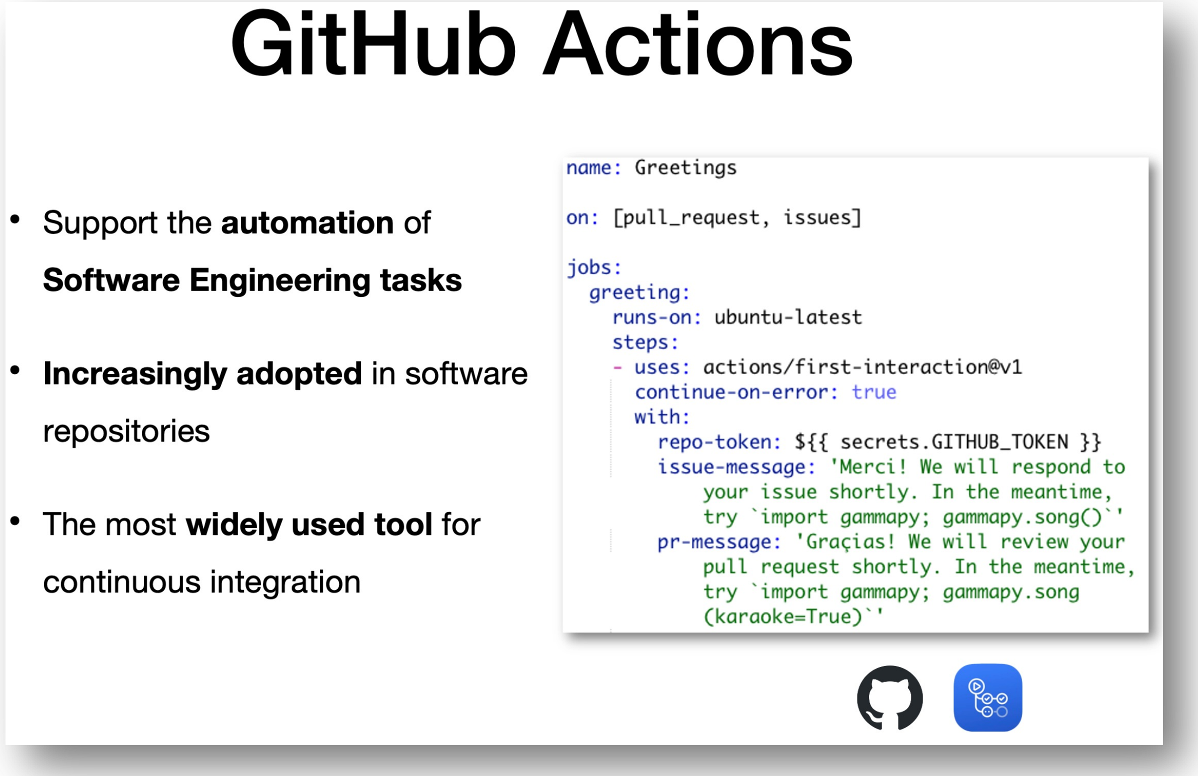
The previous example offers a Sticky Groups view. The first group of this view has 38 commits. Since the list is shown in reverse order, we select the last group.

```
stickyGroup := history stickyCommits last
```

We save the last result in the `WEHistoryExamples>>#pythonPackageExample`

- Assign an onboarding technical mentor
- Schedule regular 1:1 meetings
- Encourage using GT key resources like Lepiter notebooks, and the explorable design

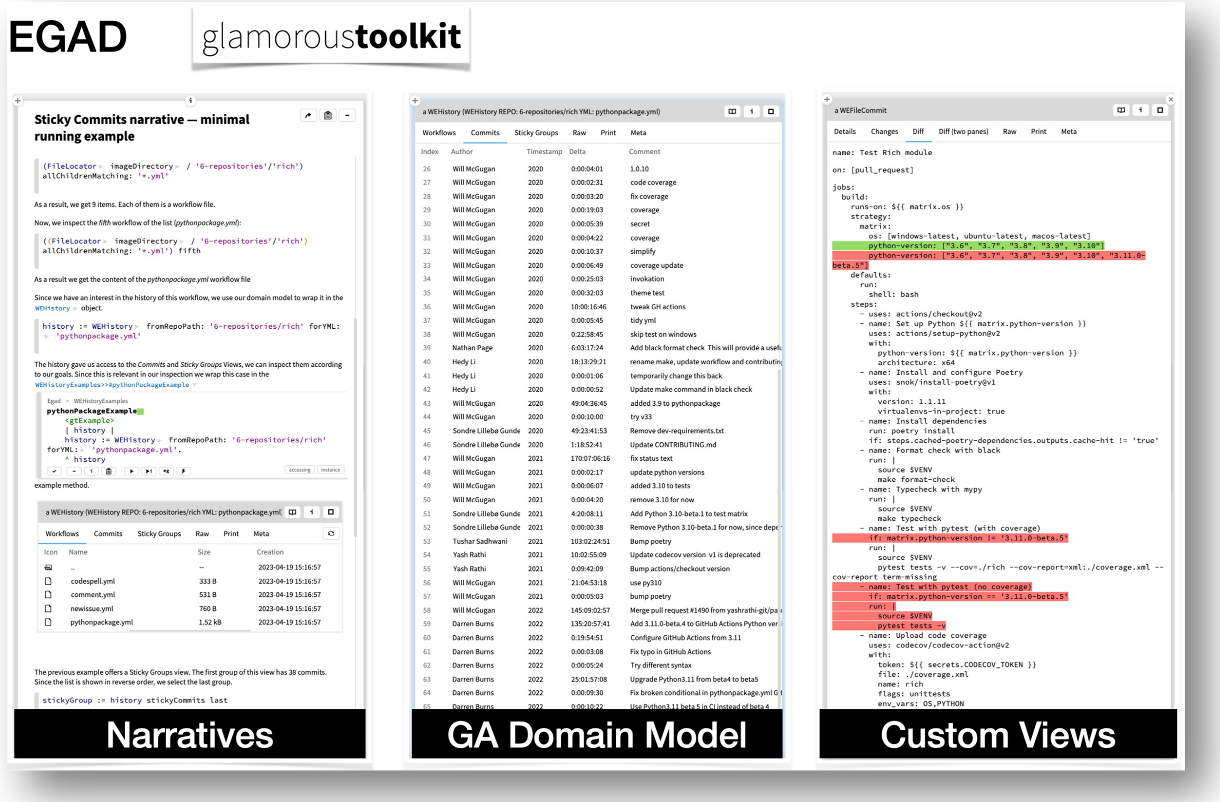
Summary



GitHub Actions

- Support the **automation** of **Software Engineering tasks**
- **Increasingly adopted** in software repositories
- The most **widely used tool** for continuous integration

```
name: Greetings

on: [pull_request, issues]

jobs:
  greeting:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/first-interaction@v1
      continue-on-error: true
      with:
        repo-token: ${{ secrets.GITHUB_TOKEN }}
        issue-message: 'Merci! We will respond to
            your issue shortly. In the meantime,
            try `import gammapy; gammapy.song()`'
        pr-message: 'Graçias! We will review your
            pull request shortly. In the meantime,
            try `import gammapy; gammapy.song
            (karaoke=True)`'
```



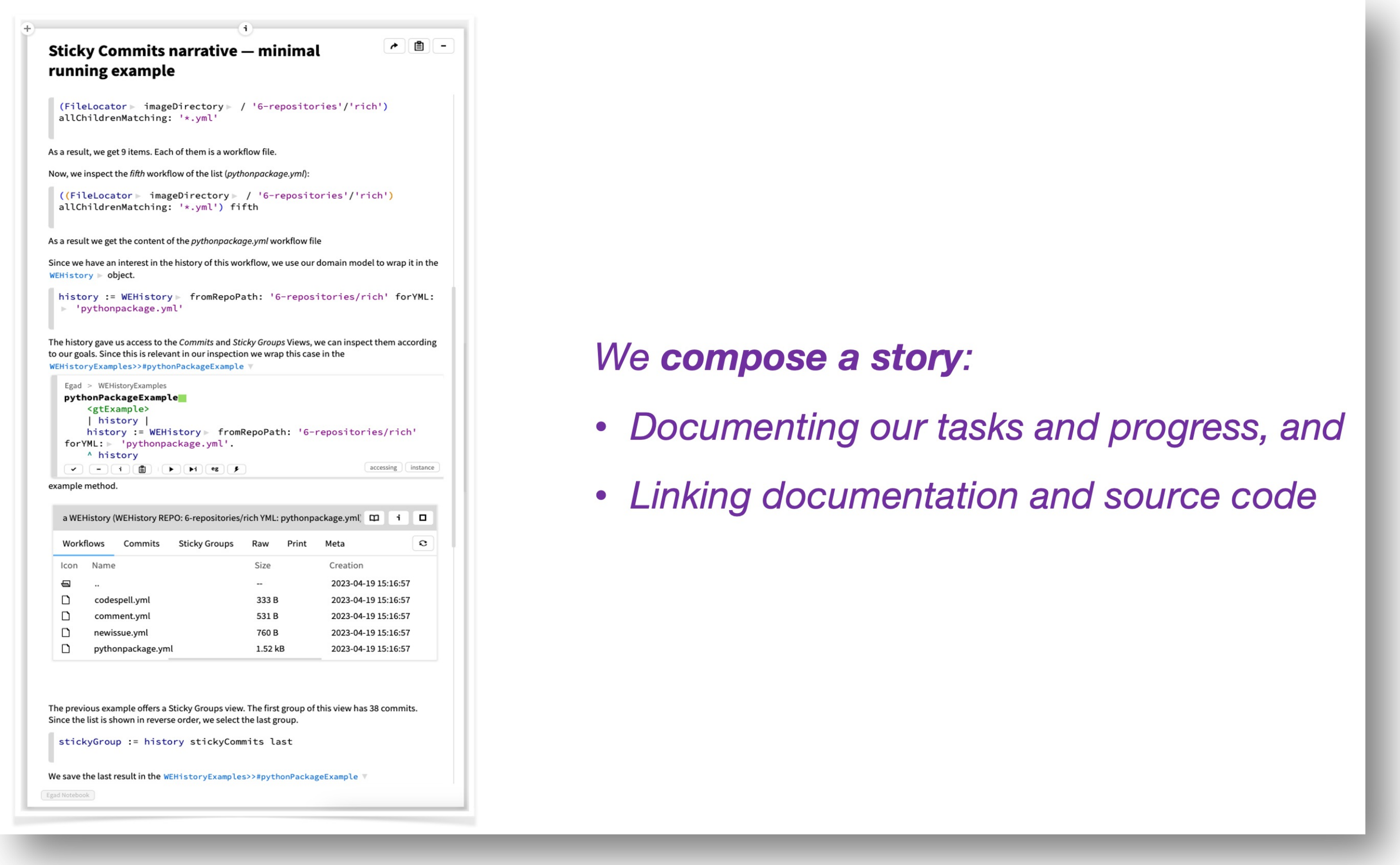
EGAD glamoroustoolkit

Sticky Commits narrative — minimal running example

Narratives

GA Domain Model

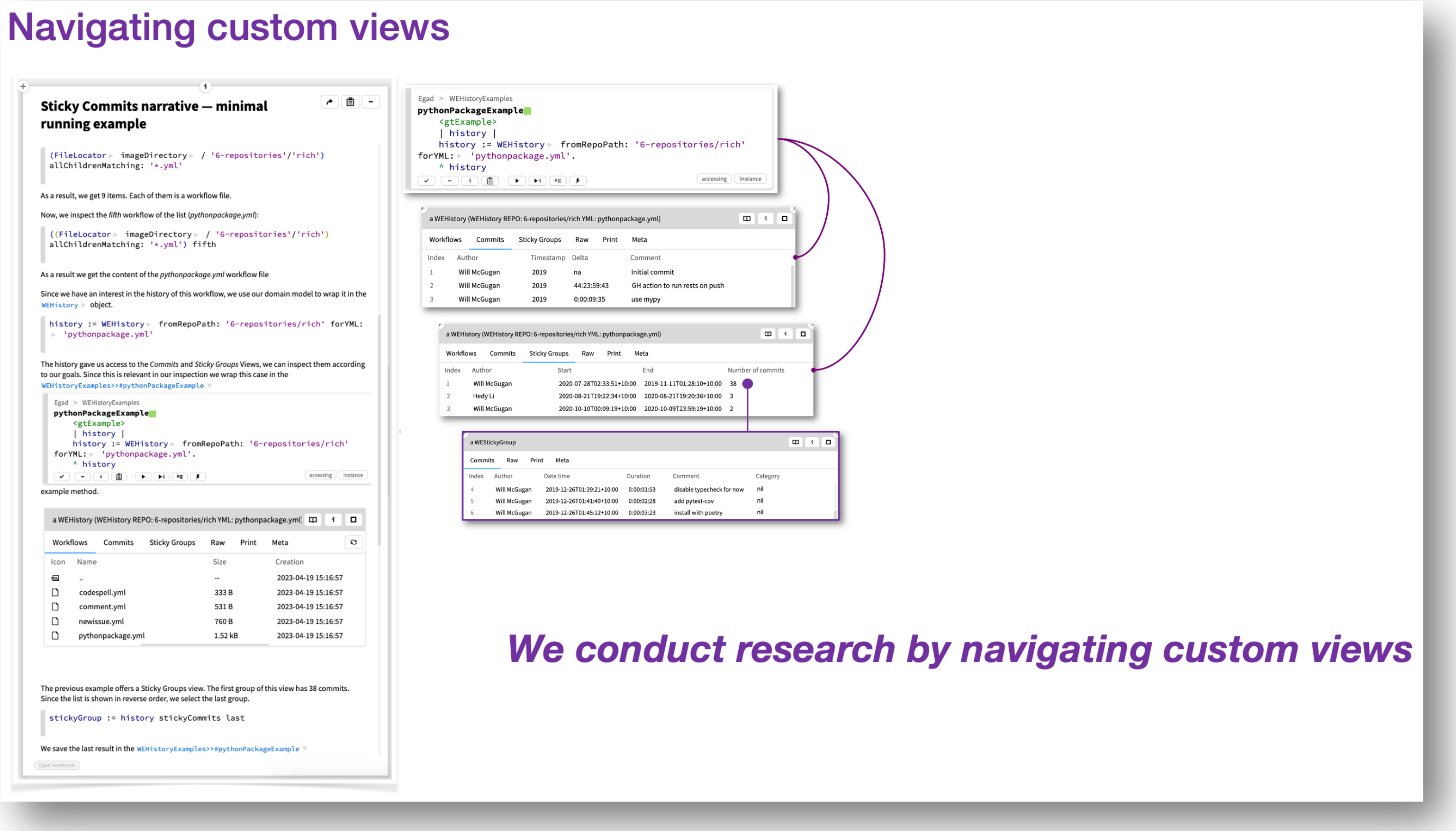
Custom Views



Sticky Commits narrative — minimal running example

We compose a story:

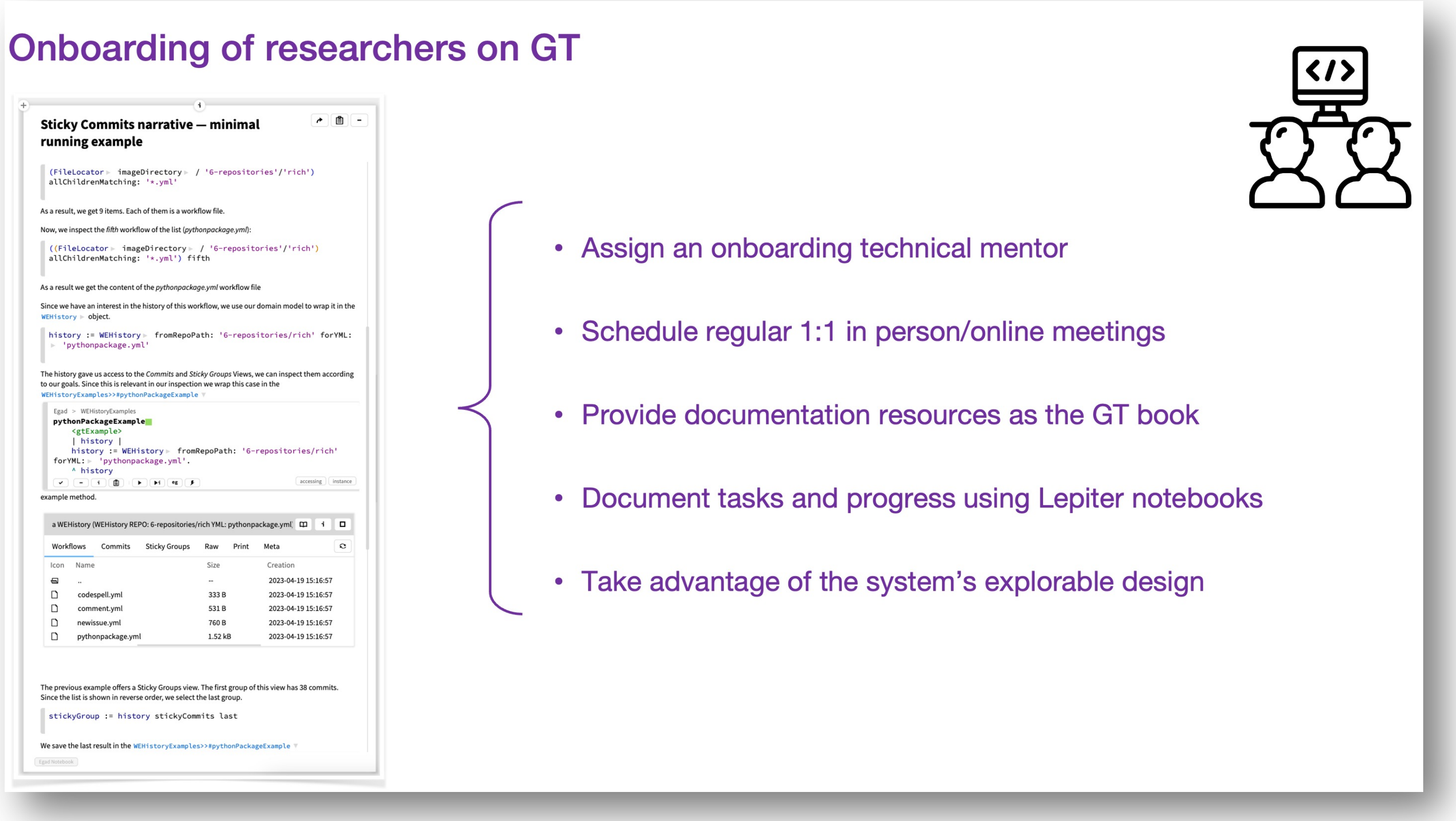
- *Documenting our tasks and progress, and*
- *Linking documentation and source code*



Navigating custom views

Sticky Commits narrative — minimal running example

We conduct research by navigating custom views



Onboarding of researchers on GT

Sticky Commits narrative — minimal running example

- Assign an onboarding technical mentor
- Schedule regular 1:1 in person/online meetings
- Provide documentation resources as the GT book
- Document tasks and progress using Lepiter notebooks
- Take advantage of the system's explorable design

Exploring GitHub Actions through EGAD: An Experience Report