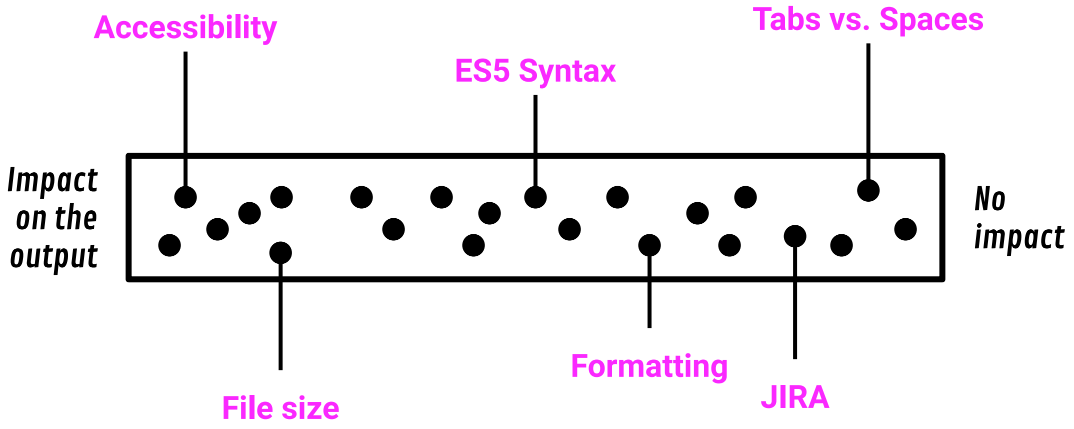
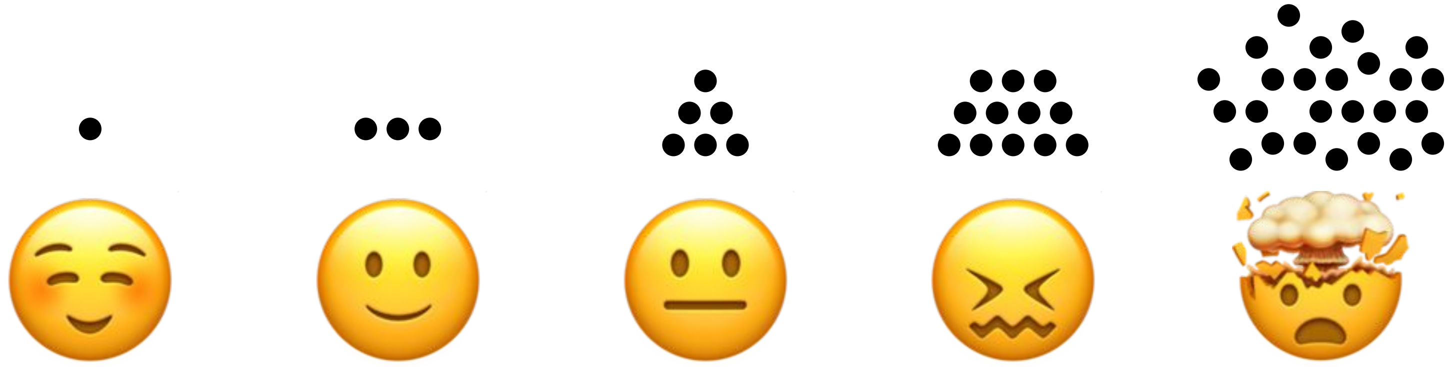
Make your tools work.

Background

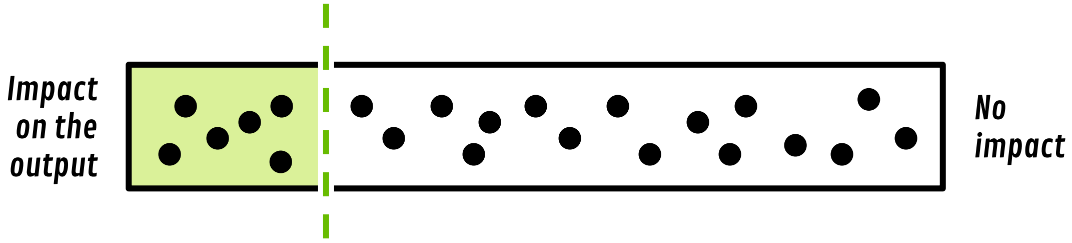
Things to do


Accessibility
ES5 Syntax
Tabs vs. Spaces
Impact on the output
No impact
File size
Formatting
JIRA

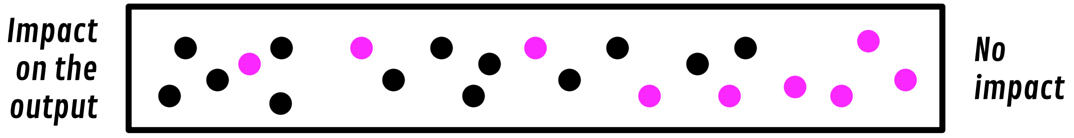
Cognitive overload



Prioritise



Automate



Let your tools do
the stuff you don't
need to do.

Make your tools work.

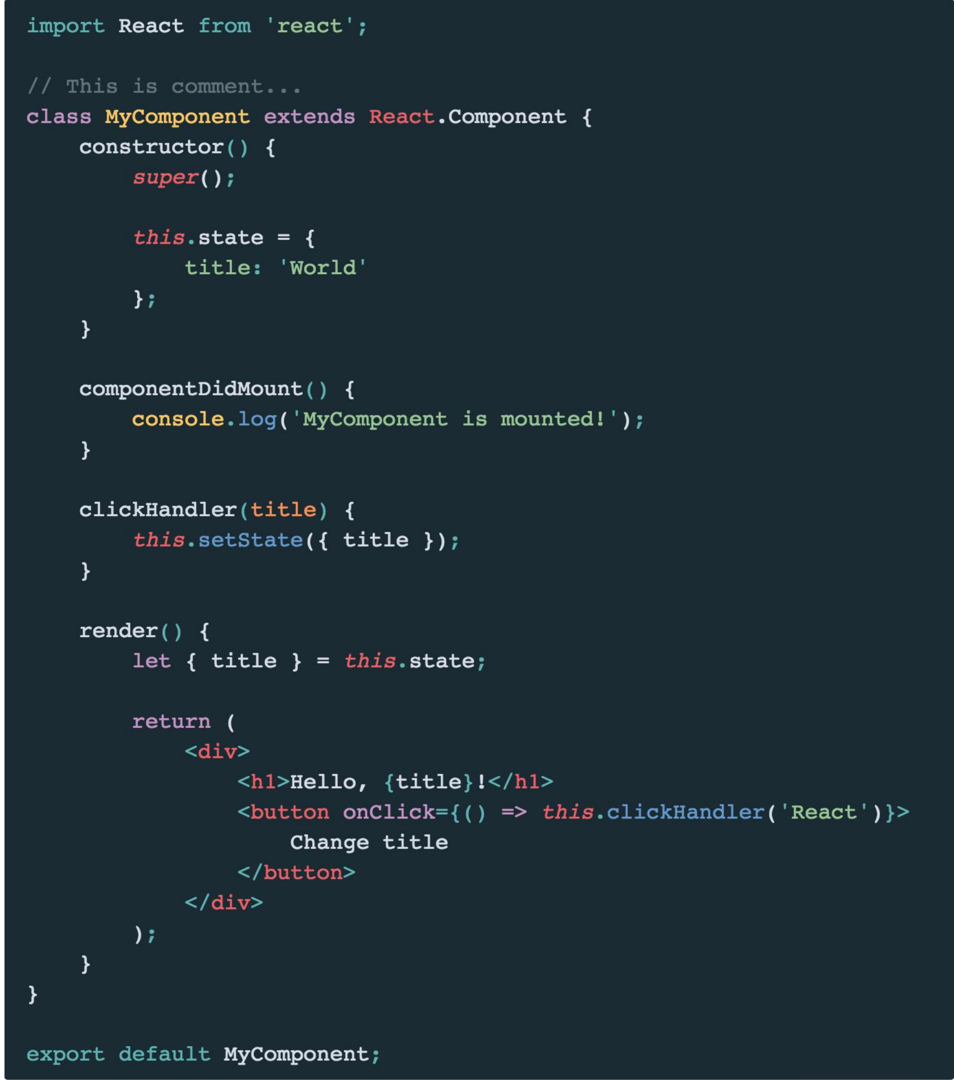
Core Principles

Our tools should:

- Help us
- Improve our output
- Be satisfying

Off the shelf tools

- We have VSCode
- Syntax highlighting
- Custom themes
- Inline syntax validation
- Integrated Git + Diffs


```
import React from 'react';

// This is comment...
class MyComponent extends React.Component {
    constructor() {
        super();

        this.state = {
            title: 'World'
        };
    }

    componentDidMount() {
        console.log('MyComponent is mounted!');
    }

    clickHandler(title) {
        this.setState({ title });
    }

    render() {
        let { title } = this.state;

        return (
            <div>
                <h1>Hello, {title}!</h1>
                <button onClick={() => this.clickHandler('React')}>
                    Change title
                </button>
            </div>
        );
    }
}

export default MyComponent;
```

Automated Formatting

- VSCode plugins and ‘Format on Save’
- Automated, not enforced

Inline Best Practice hints

- ESLint plugins:
    - react-a11y (accessibility best practices)
    - jest/recommended
    - airbnb/recommended
- As-you-type, not afterwards or one-off audit tasks

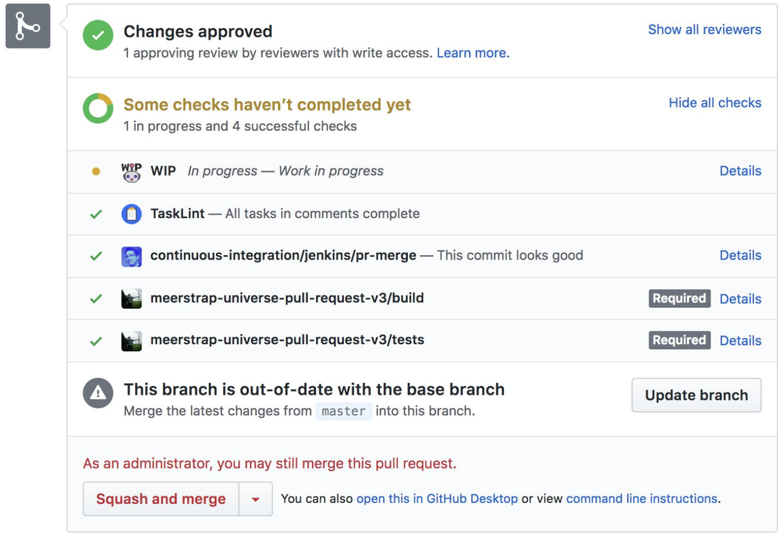
Inline / Integrated Unit Testing

- VSCode plugin
- Continual background testing
- Editor panel integration
- Inline test results, code coverage & error display

+

GitHub Integration for Jenkins / GoCD

- Automated Unit Tests, code checks in Docker
- Report results in GitHub
- Test for each commit, branch, PR



Changes approved
1 approving review by reviewers with write access. Learn more.

Show all reviewers

Some checks haven't completed yet
1 in progress and 4 successful checks

Hide all checks

●	**WIP** *In progress — Work in progress*	Details
✓	**TaskLint** — All tasks in comments complete	
✓	**continuous-integration/jenkins/pr-merge** — This commit looks good	Details
✓	**meerstrap-universe-pull-request-v3/build**	Required Details
✓	**meerstrap-universe-pull-request-v3/tests**	Required Details

This branch is out-of-date with the base branch
Merge the latest changes from `master` into this branch.

Update branch

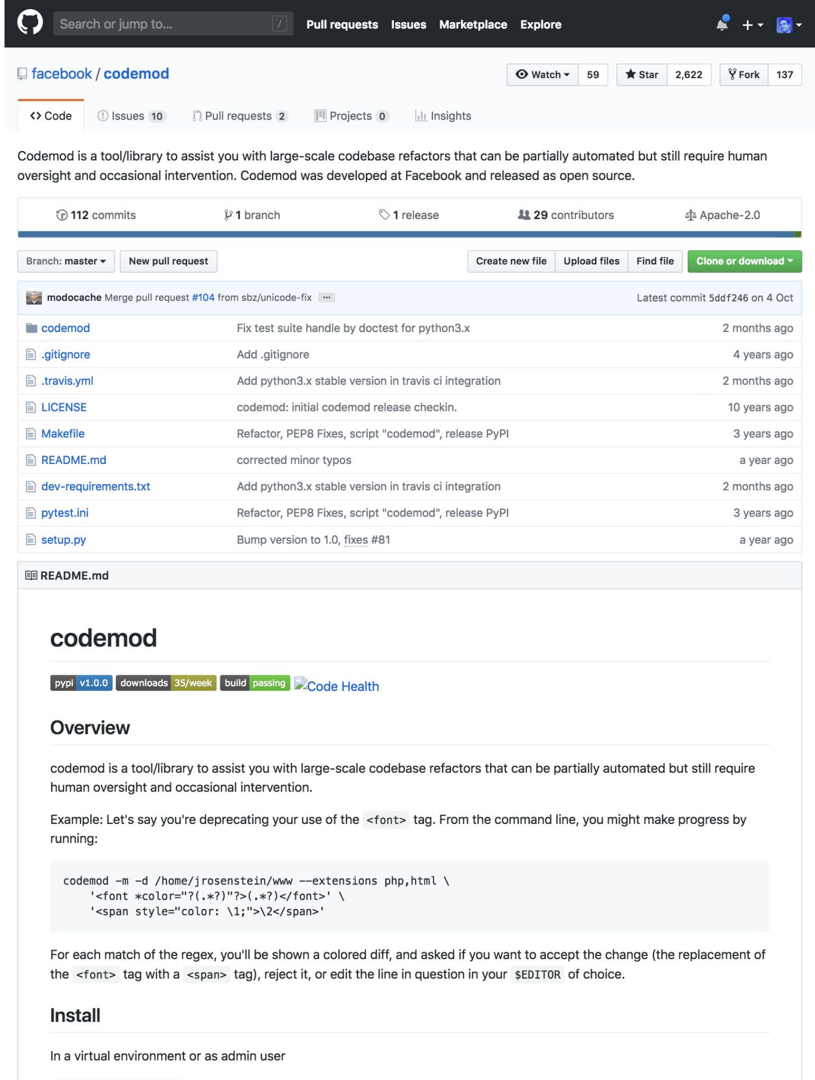
As an administrator, you may still merge this pull request.

Squash and merge

You can also open this in GitHub Desktop or view command line instructions.

Automated Refactoring

- Command-line refactoring
  - codemod
  - jscodeshift
- Inline refactoring



facebook / codemod

Codemod is a tool/library to assist you with large-scale codebase refactors that can be partially automated but still require human oversight and occasional intervention. Codemod was developed at Facebook and released as open source.

112 commits · 1 branch · 1 release · 29 contributors · Apache-2.0

modocache Merge pull request #104 from sbz/unicode-fix … Latest commit 5ddf246 on 4 Oct

codemod	Fix test suite handle by doctest for python3.x	2 months ago
.gitignore	Add .gitignore	4 years ago
.travis.yml	Add python3.x stable version in travis ci integration	2 months ago
LICENSE	codemod: initial codemod release checkin.	10 years ago
Makefile	Refactor, PEP8 Fixes, script "codemod", release PyPI	3 years ago
README.md	corrected minor typos	a year ago
dev-requirements.txt	Add python3.x stable version in travis ci integration	2 months ago
pytest.ini	Refactor, PEP8 Fixes, script "codemod", release PyPI	3 years ago
setup.py	Bump version to 1.0, fixes #81	a year ago

README.md

codemod

pypi v1.0.0 · downloads 35/week · build passing · Code Health

Overview

codemod is a tool/library to assist you with large-scale codebase refactors that can be partially automated but still require human oversight and occasional intervention.

Example: Let's say you're deprecating your use of the `<font>` tag. From the command line, you might make progress by running:

```
codemod -m -d /home/jrosenstein/www --extensions php,html \
    '<font *color="?(.*?)"?>(.*?)</font>' \
    '<span style="color: \1;">\2</span>'
```

For each match of the regex, you'll be shown a colored diff, and asked if you want to accept the change (the replacement of the `<font>` tag with a `<span>` tag), reject it, or edit the line in question in your `$EDITOR` of choice.

Install

In a virtual environment or as admin user

Build your own tools

Automated deployment scripts

- Automated repetitive steps:
  - Preflight checks (clean workspace)
  - Build packages
  - Test packages
  - Regenerate docs
  - Tag in git
  - Publish to npm

GitHub-JIRA integration

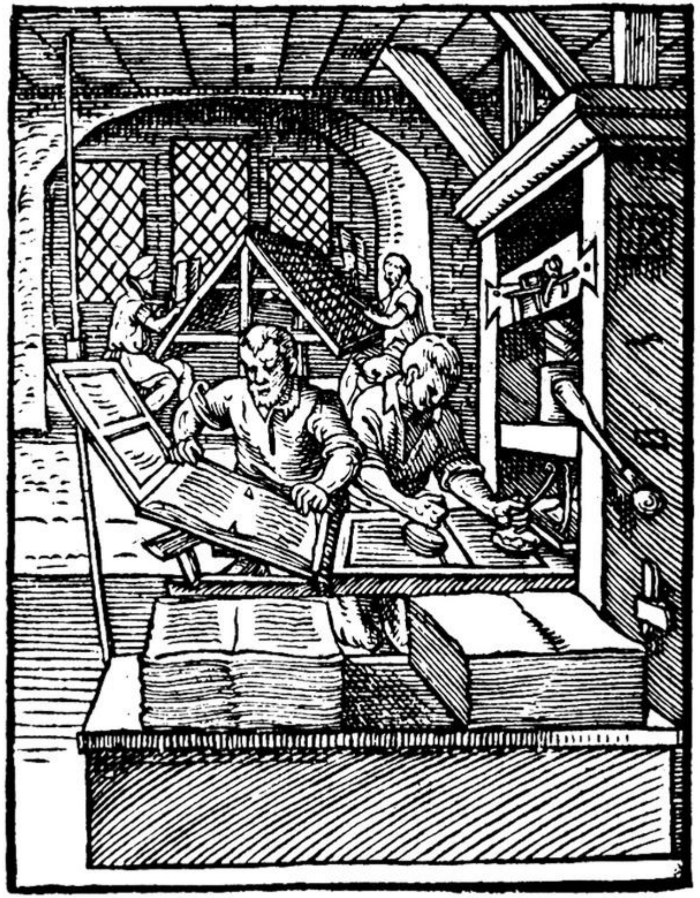
- Creates tickets in JIRA for GitHub Issues & Pull Requests
- Transition tickets in reaction to GitHub webhook events
- Assign users, mirror comments & updates

Craftsmanship

Software development is a comparatively new profession.

Building your own tools is not a new idea - it's an old one, actually.

Mass produced tools work for common tasks, but for in-depth bespoke work, it's worthwhile investing in custom tools.



Summary

Make your tools work for you.