Fighting chaos in a monorepo

Monorepo is a good servant, but a bad master productboard

Jakub Beneš

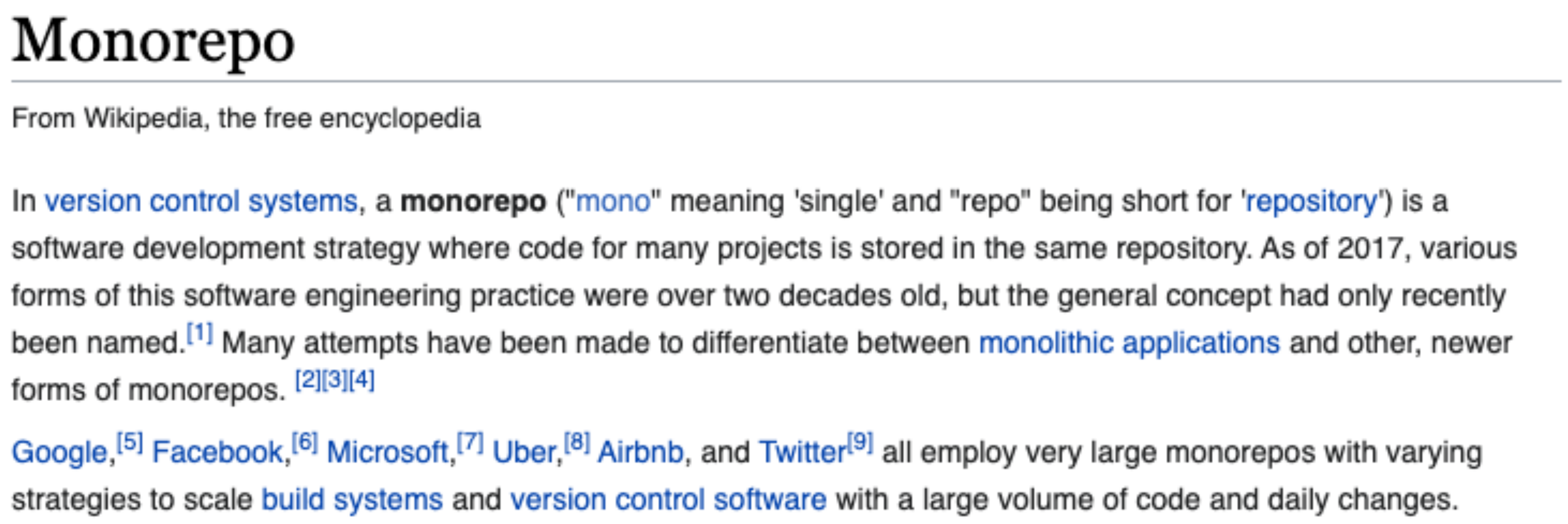
Engineering Manager @ Productboard

@jukben @jukben https://jukben.codes

Agenda

- What is monorepo
- What problems we faced at Productboard
- What strategies we have deployed
- Takeaways

What's monorepo



Monorepo

From Wikipedia, the free encyclopedia

In version control systems, a **monorepo** ("mono" meaning 'single' and "repo" being short for 'repository') is a software development strategy where code for many projects is stored in the same repository. As of 2017, various forms of this software engineering practice were over two decades old, but the general concept had only recently been named.[1] Many attempts have been made to differentiate between monolithic applications and other, newer forms of monorepos. [2][3][4]

Google,[5] Facebook,[6] Microsoft,[7] Uber,[8] Airbnb, and Twitter[9] all employ very large monorepos with varying strategies to scale build systems and version control software with a large volume of code and daily changes.

Why and why not?

Pros

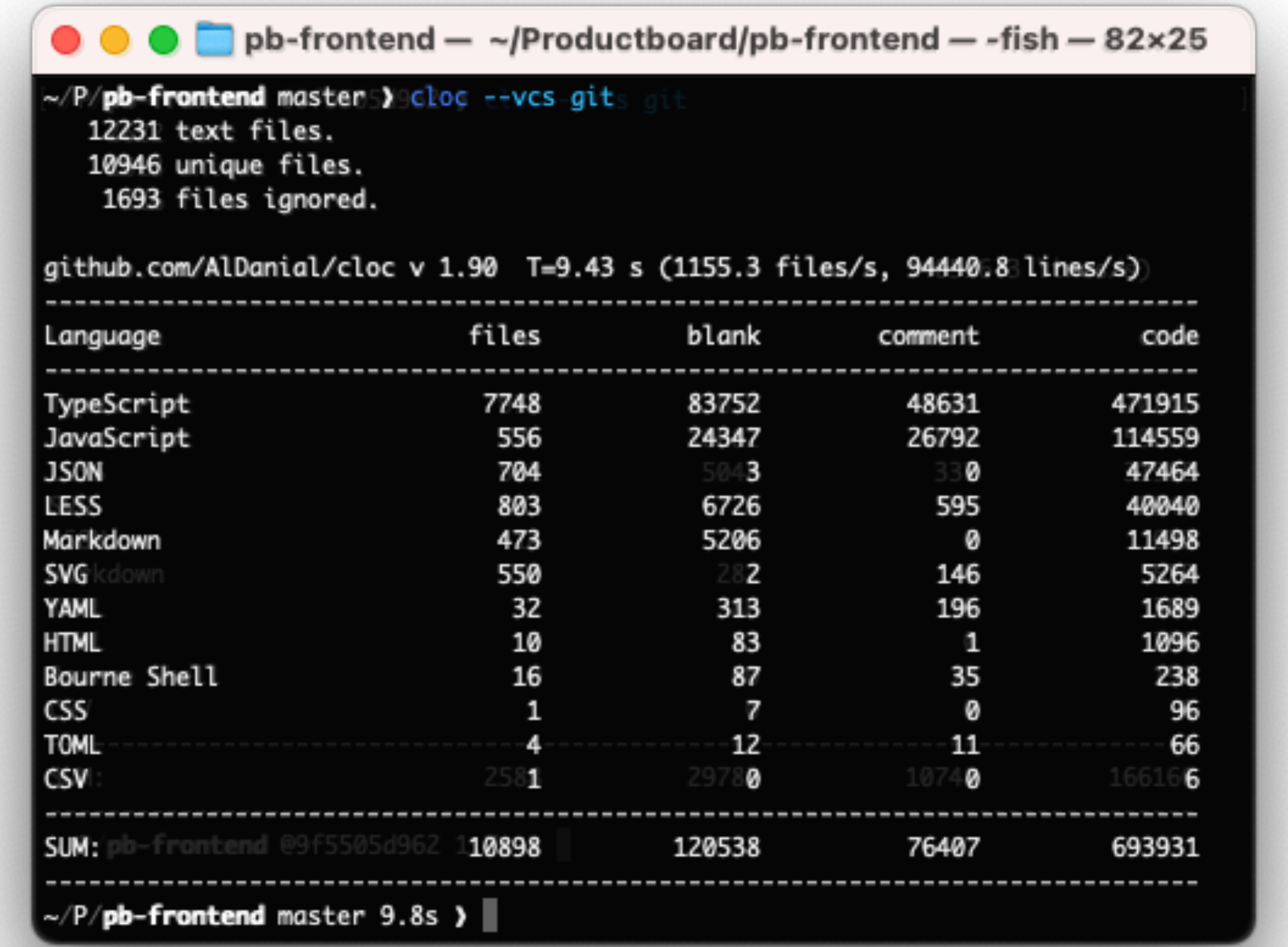
- Better visibility and collaboration across teams
- Simplified dependency management
- Easier large scale refactoring

Cons

- Build pipelines
- VSC Tooling Challenges
- Limitations Around Access Control

FLAME WARS

Throwback



```
~/P/pb-frontend master ❯ cloc --vcs git
   12231 text files.
   10946 unique files.
    1693 files ignored.

github.com/AlDanial/cloc v 1.90  T=9.43 s (1155.3 files/s, 94440.8 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
TypeScript                    7748          83752          48631         471915
JavaScript                     556          24347          26792         114559
JSON                           704              3              0          47464
LESS                           803           6726            595          40040
Markdown                       473           5206              0          11498
SVG                            550              2            146           5264
YAML                            32            313            196           1689
HTML                            10             83              1           1096
Bourne Shell                    16             87             35            238
CSS                              1              7              0             96
TOML                             4             12             11             66
CSV                              1              0              0              6
-------------------------------------------------------------------------------
SUM:                         10898         120538          76407         693931
-------------------------------------------------------------------------------
~/P/pb-frontend master 9.8s ❯
```

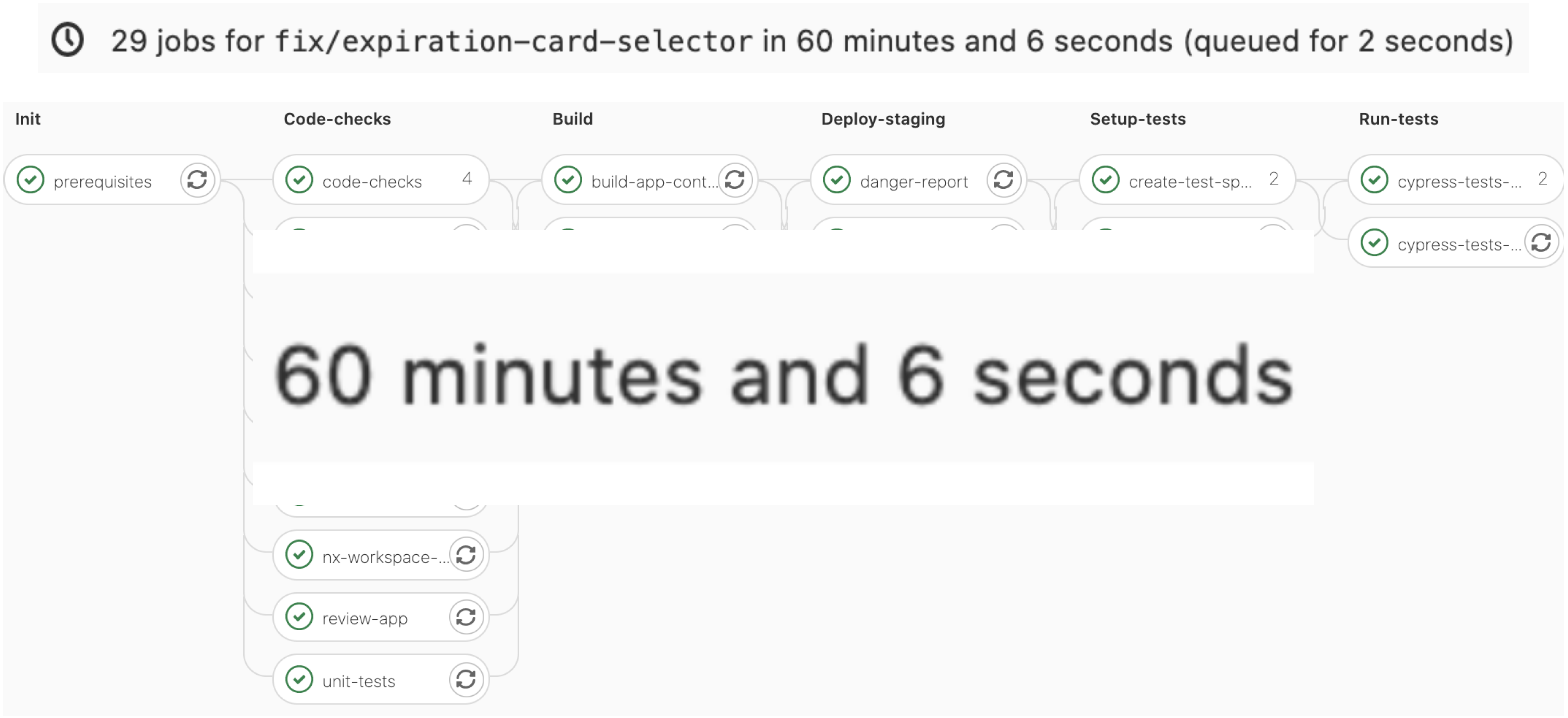
85%

bigger

Throwback

circleci $\rightarrow$ CI/CD

CI/CD Pipeline



29 jobs for fix/expiration-card-selector in 60 minutes and 6 seconds (queued for 2 seconds)

Init	Code-checks	Build	Deploy-staging	Setup-tests	Run-tests
✓ prerequisites	✓ code-checks 4	✓ build-app-cont...	✓ danger-report	✓ create-test-sp... 2	✓ cypress-tests-... 2
					✓ cypress-tests-...
	✓ nx-workspace-...				
	✓ review-app				
	✓ unit-tests				

60 minutes and 6 seconds

It doesn't scale
CHANGE MY MIND
imgflip.com

Tooling is our friend

- We have conducted research for tooling which would help us to maintain the monorepo better.
    - Manage a complex dependency graph
    - Build only affected projects
    - Provide API scaffold code

Tooling is our friend

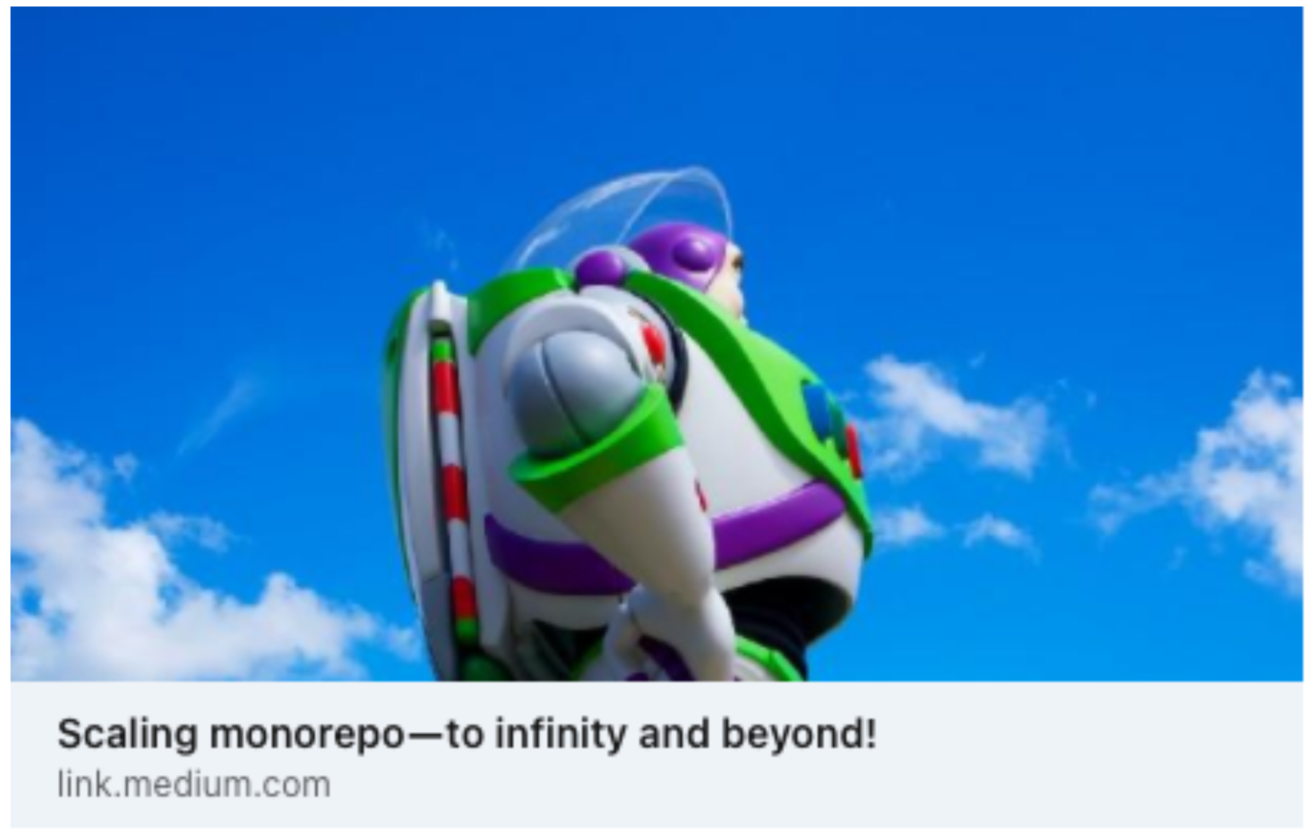
rush

Lerna

Nx

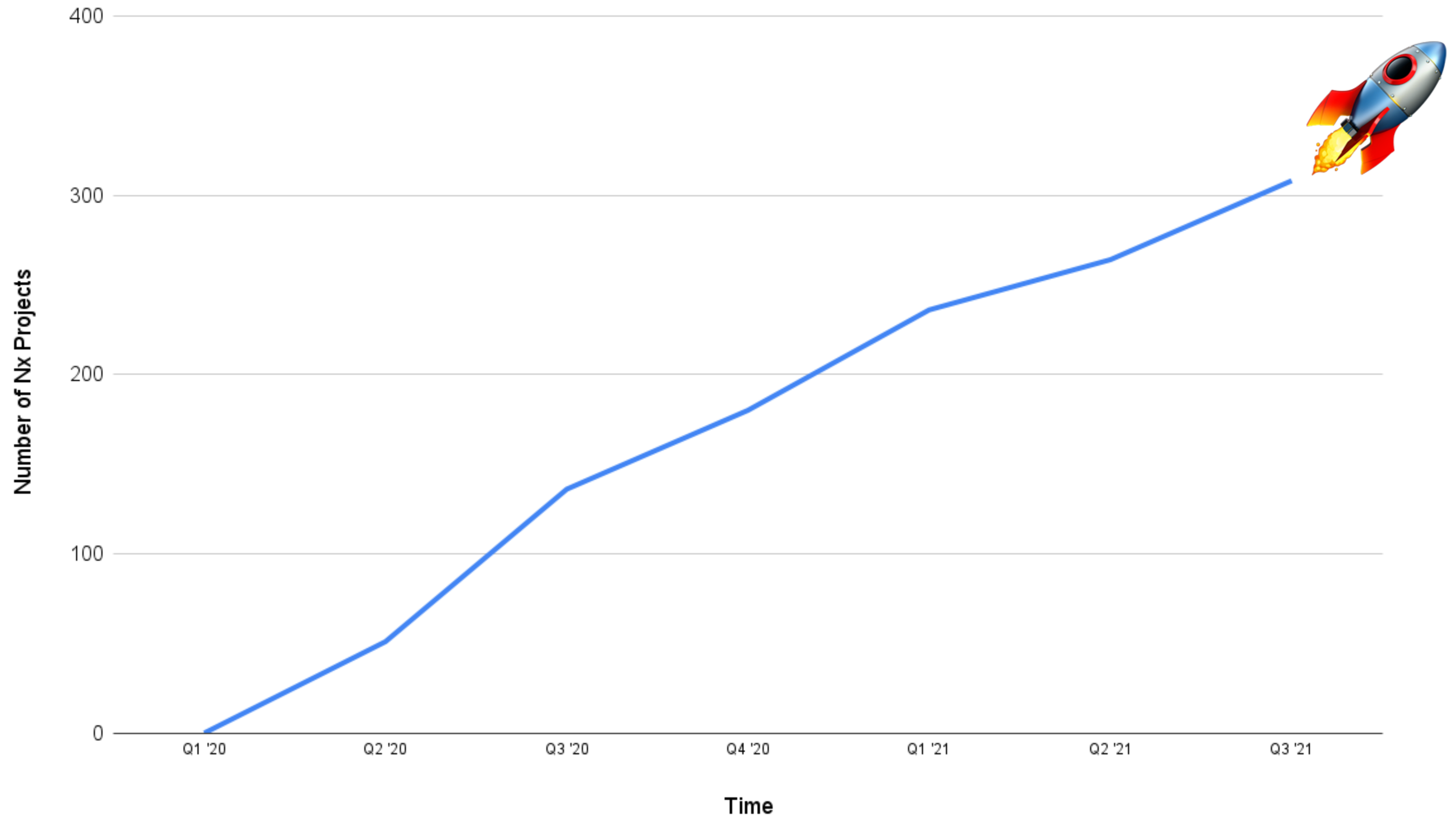
Nx proven to be right choice

- We started to break down our monolith into smaller chunks (Nx projects)
  - Possibility to run them separately (eslint, jest, build, deployment)
  - Isolation
  - Ownership


Scaling monorepo—to infinity and beyond!
link.medium.com

https://medium.com/productboard-engineering/

Adoption



400
300
200
100
0
Number of Nx Projects
Q1 '20
Q2 '20
Q3 '20
Q4 '20
Q1 '21
Q2 '21
Q3 '21
Time

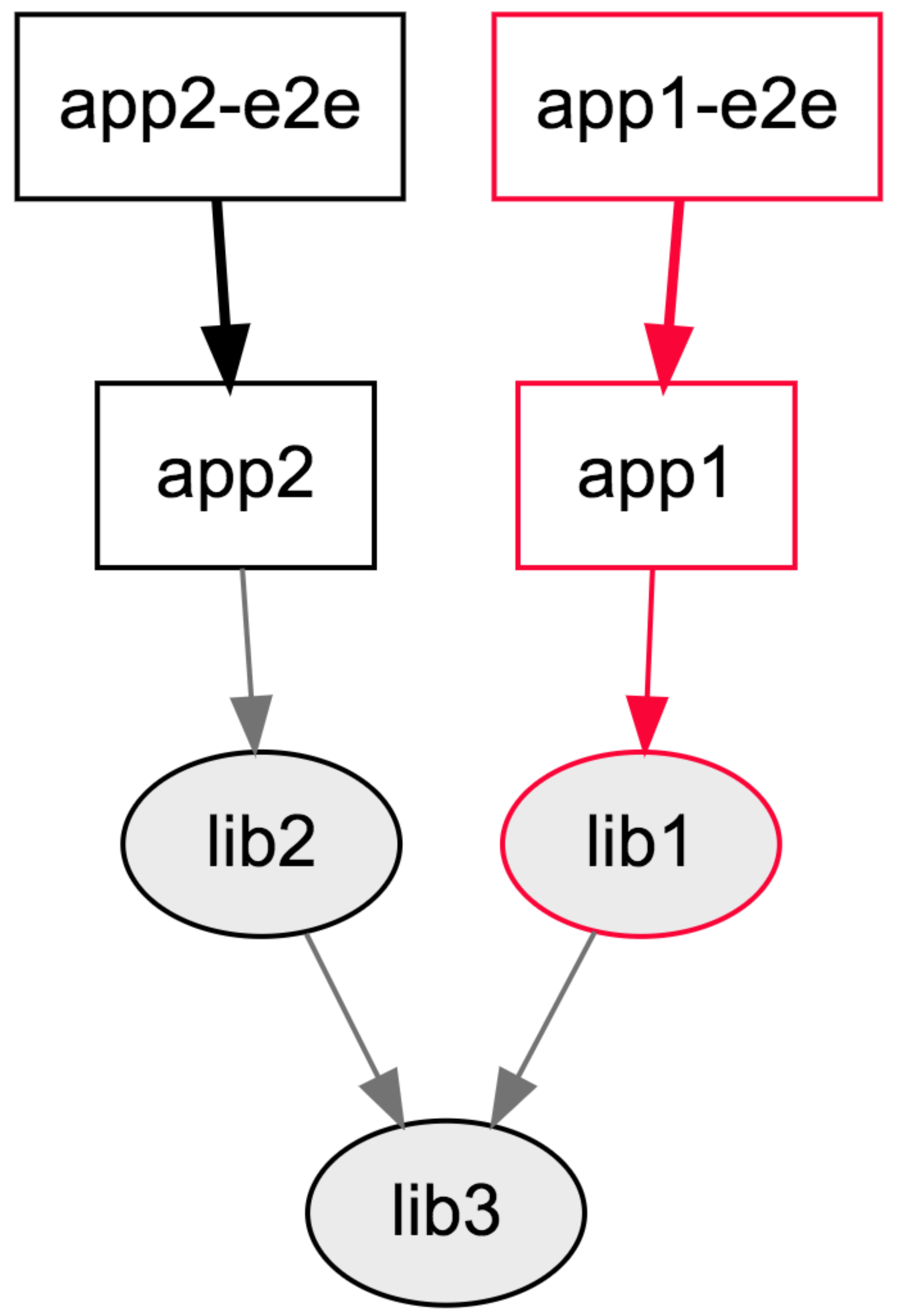
BLUE ORIGIN
BLUE ORIGIN

This is what we have got...

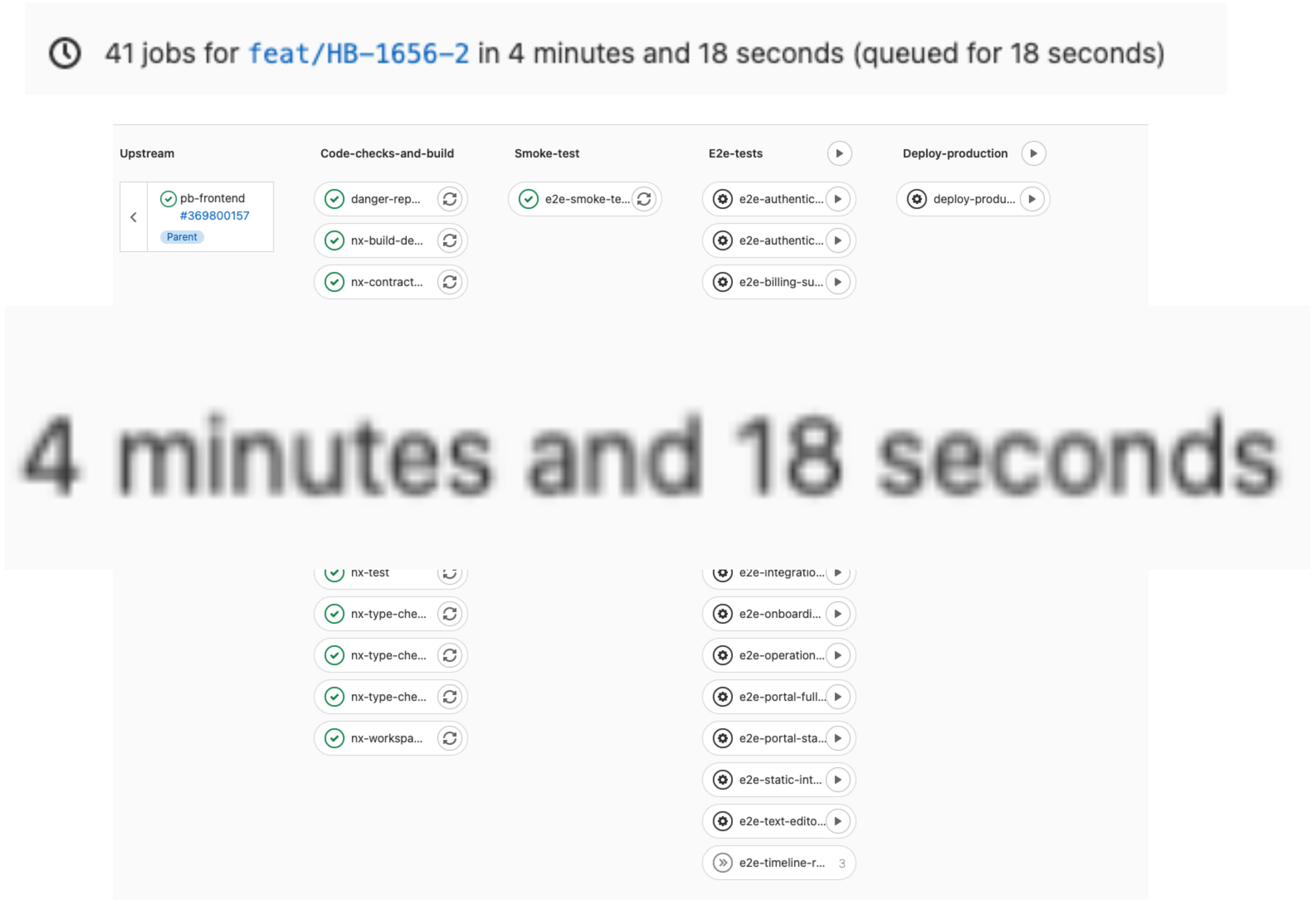
Benefit no. 1: Context Aware Pipeline

- By using `nx affected` we were able to run only that code that changed or was affected by dependency graph

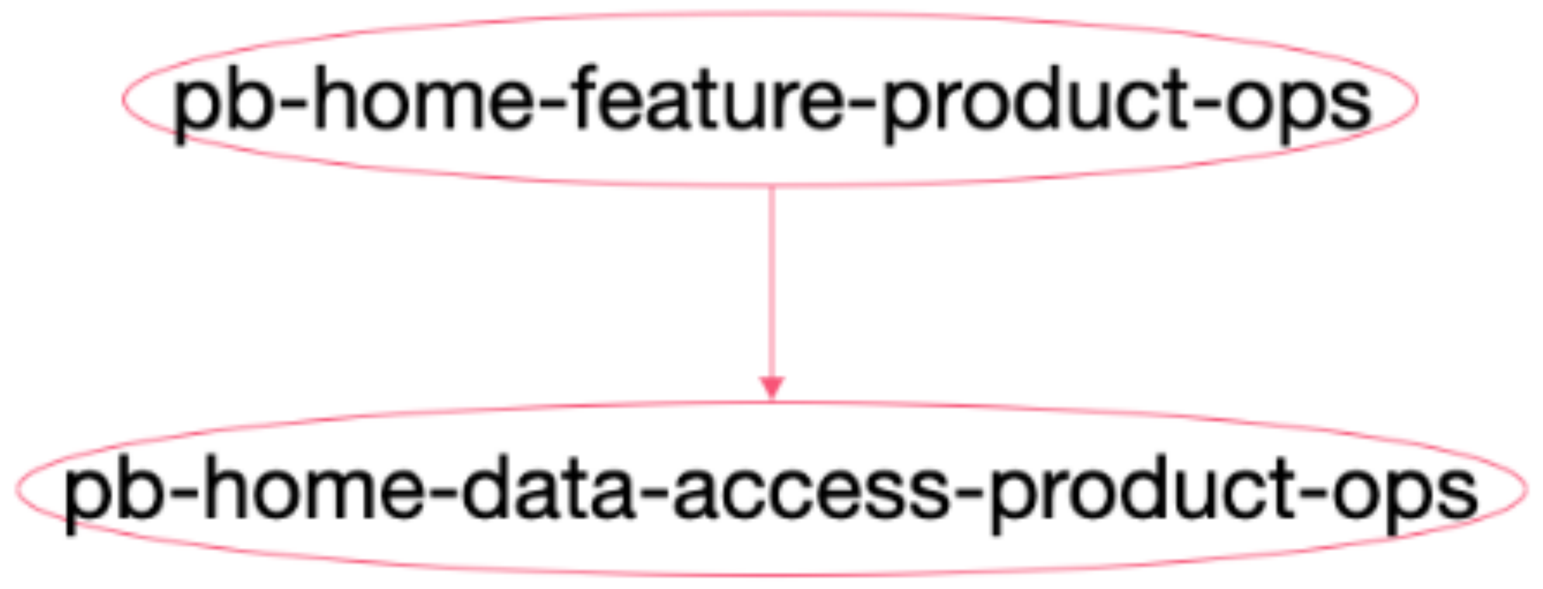
- Nx has also support for distributed caching across all environments



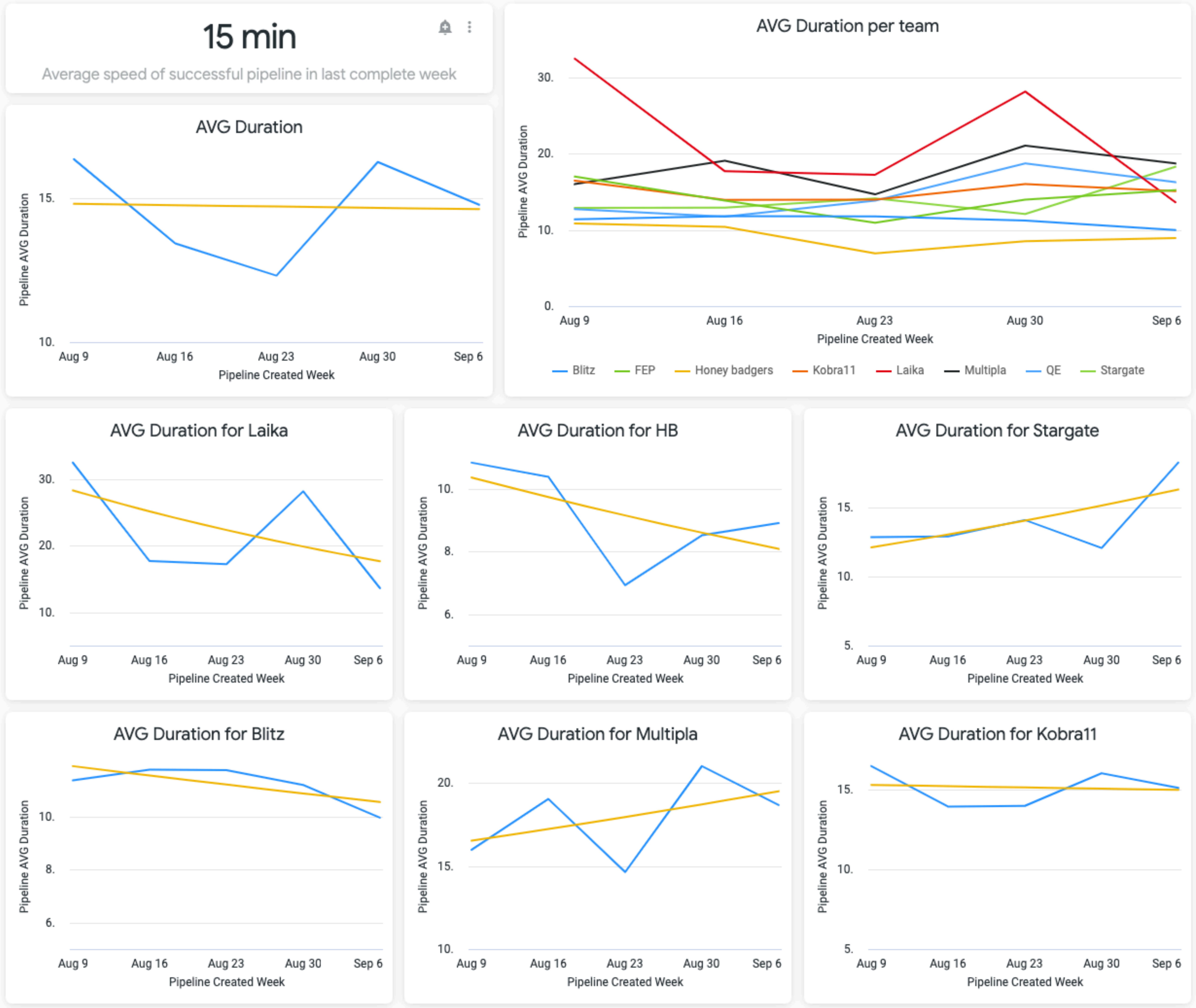
Benefit no. 1: Context Aware Pipeline


41 jobs for feat/HB-1656-2 in 4 minutes and 18 seconds (queued for 18 seconds)

4 minutes and 18 seconds



Benefit no. 1: Context Aware Pipeline

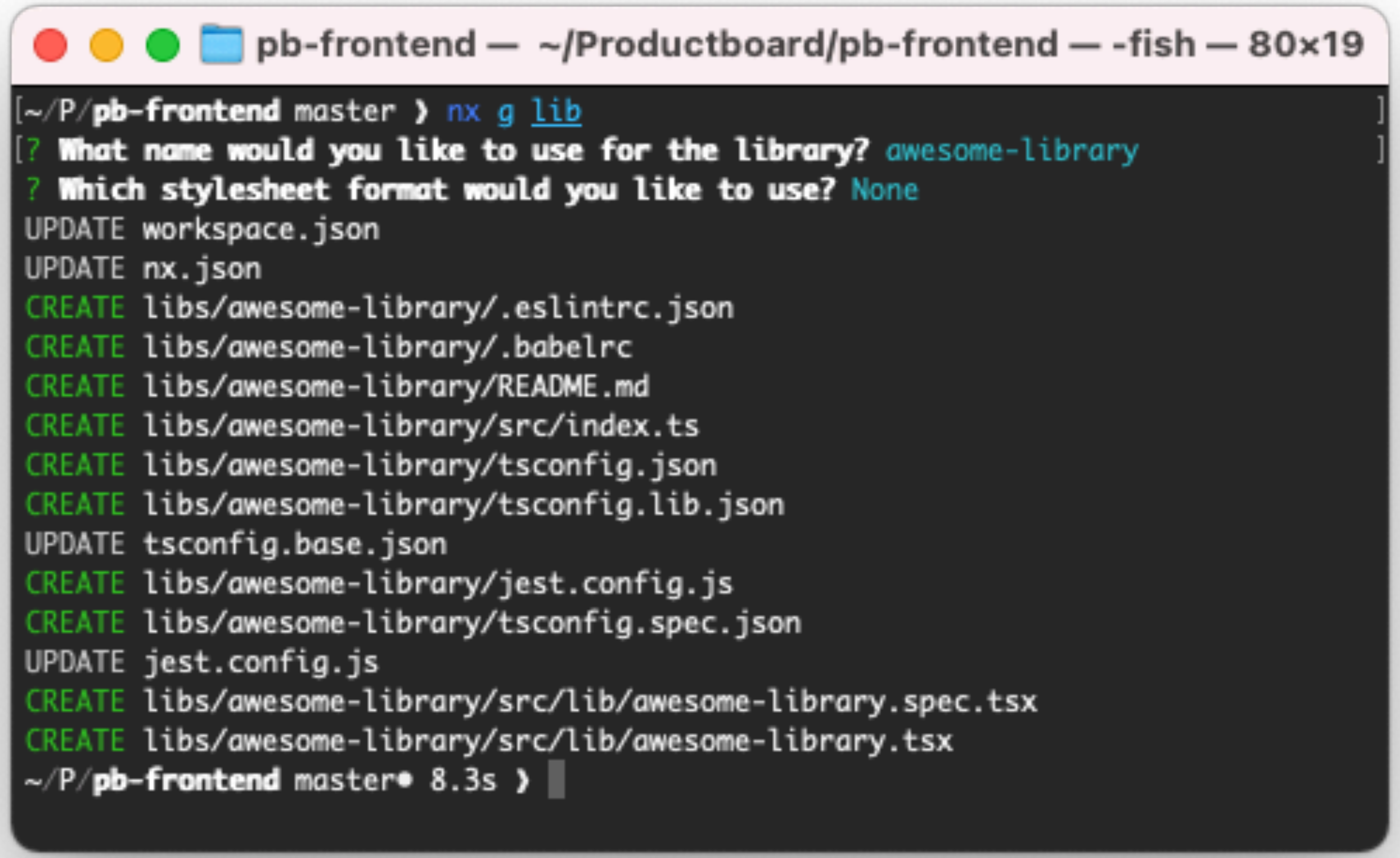


75%

faster

Benefit no. 2: Consistence and ownership

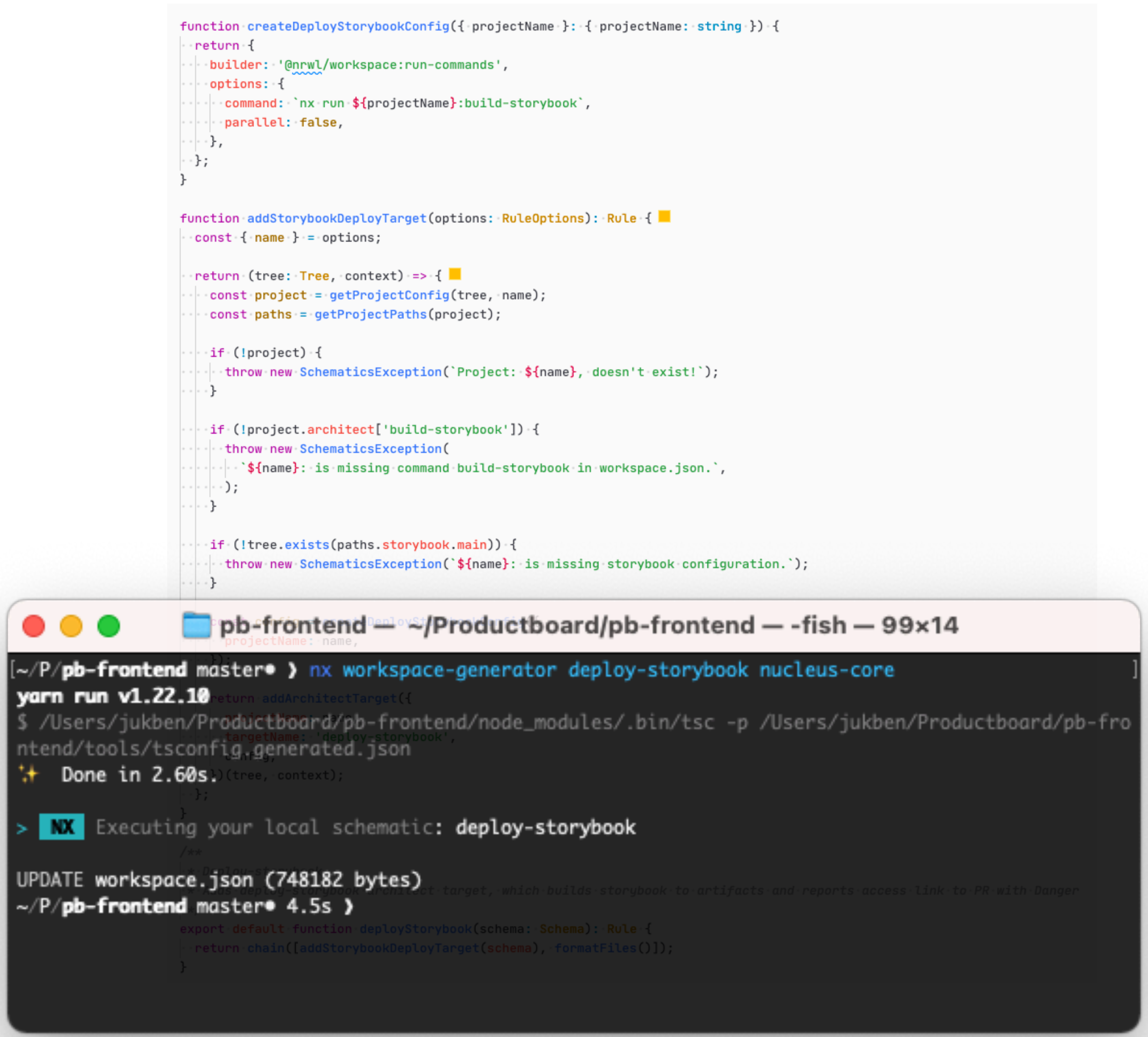
- Every lib is generated with sane default values and configuration
- By default we push author to make entry into `CODEOWNER` file



```
~/P/pb-frontend master ❯ nx g lib
? What name would you like to use for the library? awesome-library
? Which stylesheet format would you like to use? None
UPDATE workspace.json
UPDATE nx.json
CREATE libs/awesome-library/.eslintrc.json
CREATE libs/awesome-library/.babelrc
CREATE libs/awesome-library/README.md
CREATE libs/awesome-library/src/index.ts
CREATE libs/awesome-library/tsconfig.json
CREATE libs/awesome-library/tsconfig.lib.json
UPDATE tsconfig.base.json
CREATE libs/awesome-library/jest.config.js
CREATE libs/awesome-library/tsconfig.spec.json
UPDATE jest.config.js
CREATE libs/awesome-library/src/lib/awesome-library.spec.tsx
CREATE libs/awesome-library/src/lib/awesome-library.tsx
~/P/pb-frontend master• 8.3s ❯
```

Benefit no. 3: Migration framework

- Nx has support of "generators" to scaffold tests, components and file structure in general
  - Comes with opinionated structure but it's extensible



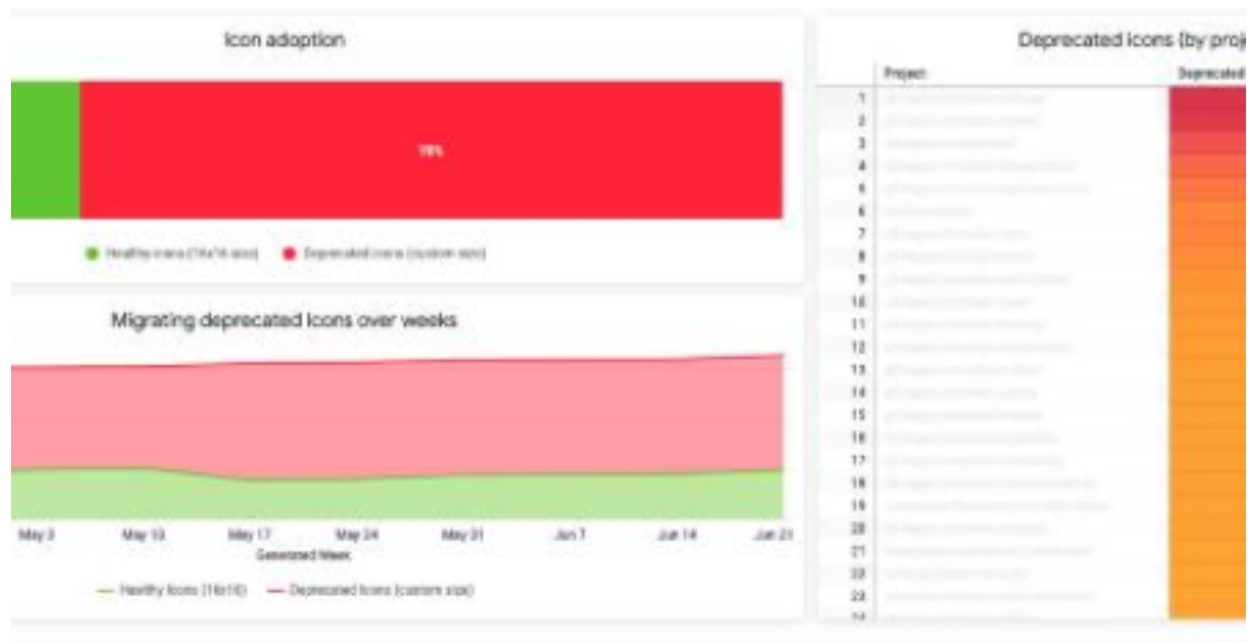
```ts
function createDeployStorybookConfig({ projectName }: { projectName: string }) {
  return {
    builder: '@nrwl/workspace:run-commands',
    options: {
      command: `nx run ${projectName}:build-storybook`,
      parallel: false,
    },
  };
}

function addStorybookDeployTarget(options: RuleOptions): Rule {
  const { name } = options;

  return (tree: Tree, context) => {
    const project = getProjectConfig(tree, name);
    const paths = getProjectPaths(project);

    if (!project) {
      throw new SchematicsException(`Project: ${name}, doesn't exist!`);
    }

    if (!project.architect['build-storybook']) {
      throw new SchematicsException(
        `${name}: is missing command build-storybook in workspace.json.`,
      );
    }

    if (!tree.exists(paths.storybook.main)) {
      throw new SchematicsException(`${name}: is missing storybook configuration.`);
    }
```

pb-frontend — ~/Productboard/pb-frontend — -fish — 99×14

```
~/P/pb-frontend master• ❯ nx workspace-generator deploy-storybook nucleus-core
yarn run v1.22.10
$ /Users/jukben/Productboard/pb-frontend/node_modules/.bin/tsc -p /Users/jukben/Productboard/pb-fro
ntend/tools/tsconfig.generated.json
✨  Done in 2.60s.

>  NX  Executing your local schematic: deploy-storybook

UPDATE workspace.json (748182 bytes)
~/P/pb-frontend master• 4.5s ❯
```

Cherry on the top: Observability

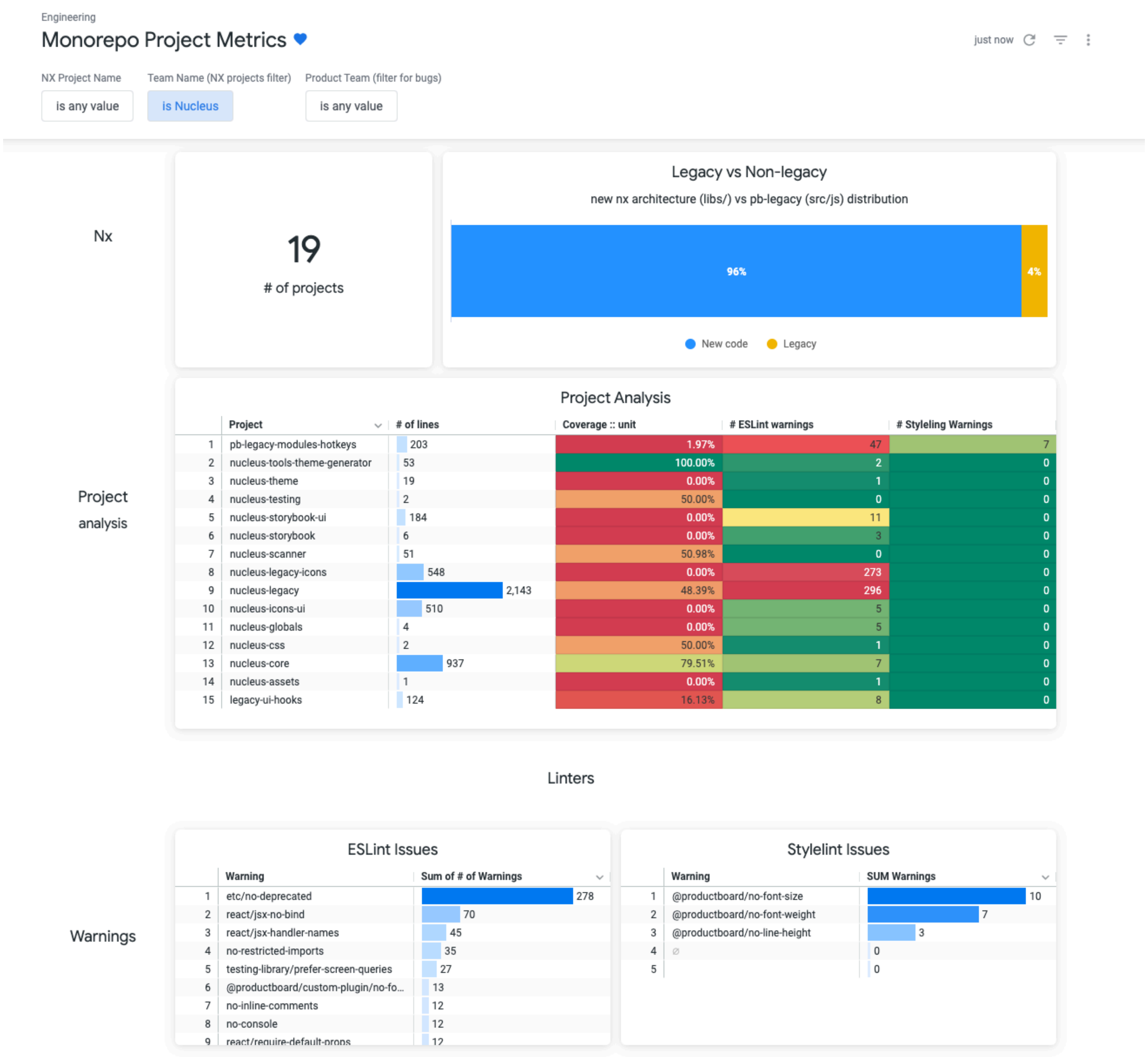
- Together with `CODEOWNERS` we are able to map Nx projects to ESlint issues, test coverage and more.



Icon adoption
Migrating deprecated icons over weeks
Deprecated icons (by proj

How we measure adoption of a design system at Productboard
link.medium.com

https://medium.com/productboard-engineering/



Monorepo Project Metrics

NX Project Name	Team Name (NX projects filter)	Product Team (filter for bugs)
is any value	is Nucleus	is any value

Project Analysis

	Project	# of lines	Coverage :: unit	# ESLint warnings	# Styleling Warnings
1	pb-legacy-modules-hotkeys	203	1.97%	47	7
2	nucleus-tools-theme-generator	53	100.00%	2	0
3	nucleus-theme	19	0.00%	1	0
4	nucleus-testing	2	50.00%	0	0
5	nucleus-storybook-ui	184	0.00%	11	0
6	nucleus-storybook	6	0.00%	3	0
7	nucleus-scanner	51	50.98%	0	0
8	nucleus-legacy-icons	548	0.00%	273	0
9	nucleus-legacy	2,143	48.39%	296	0
10	nucleus-icons-ui	510	0.00%	5	0
11	nucleus-globals	4	0.00%	5	0
12	nucleus-css	2	50.00%	1	0
13	nucleus-core	937	79.51%	7	0
14	nucleus-assets	1	0.00%	1	0
15	legacy-ui-hooks	124	16.13%	8	0

ESLint Issues

	Warning	Sum of # of Warnings
1	etc/no-deprecated	278
2	react/jsx-no-bind	70
3	react/jsx-handler-names	45
4	no-restricted-imports	35
5	testing-library/prefer-screen-queries	27
6	@productboard/custom-plugin/no-fo…	13
7	no-inline-comments	12
8	no-console	12
9	react/require-default-props	12

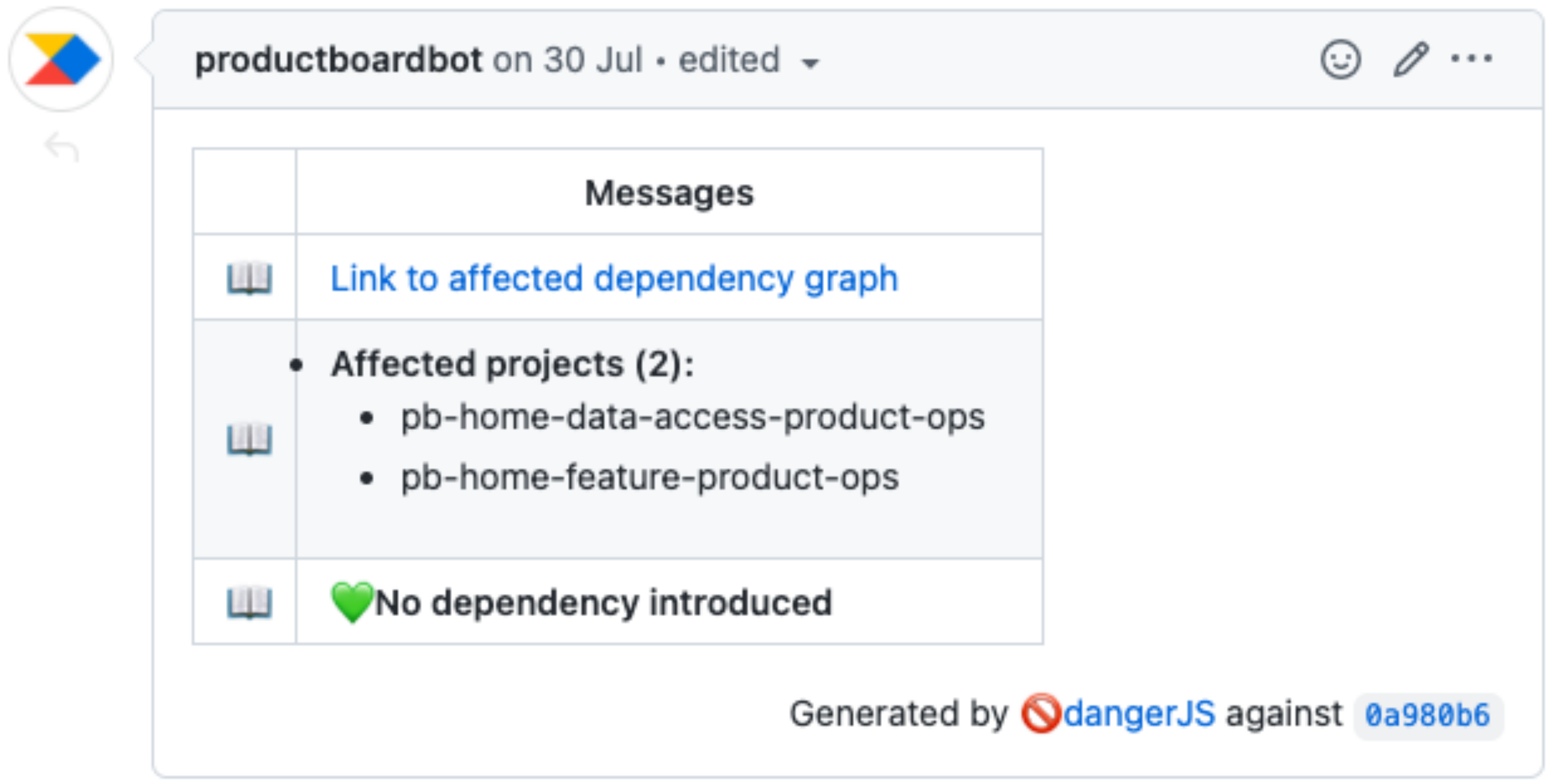
Stylelint Issues

	Warning	SUM Warnings
1	@productboard/no-font-size	10
2	@productboard/no-font-weight	7
3	@productboard/no-line-height	3
4	∅	0
5		0

Next, what else do we have...

Danger.js

- Danger runs during your CI process, and gives teams the chance to automate common code review chores.
- Make robots do chores! 🤖

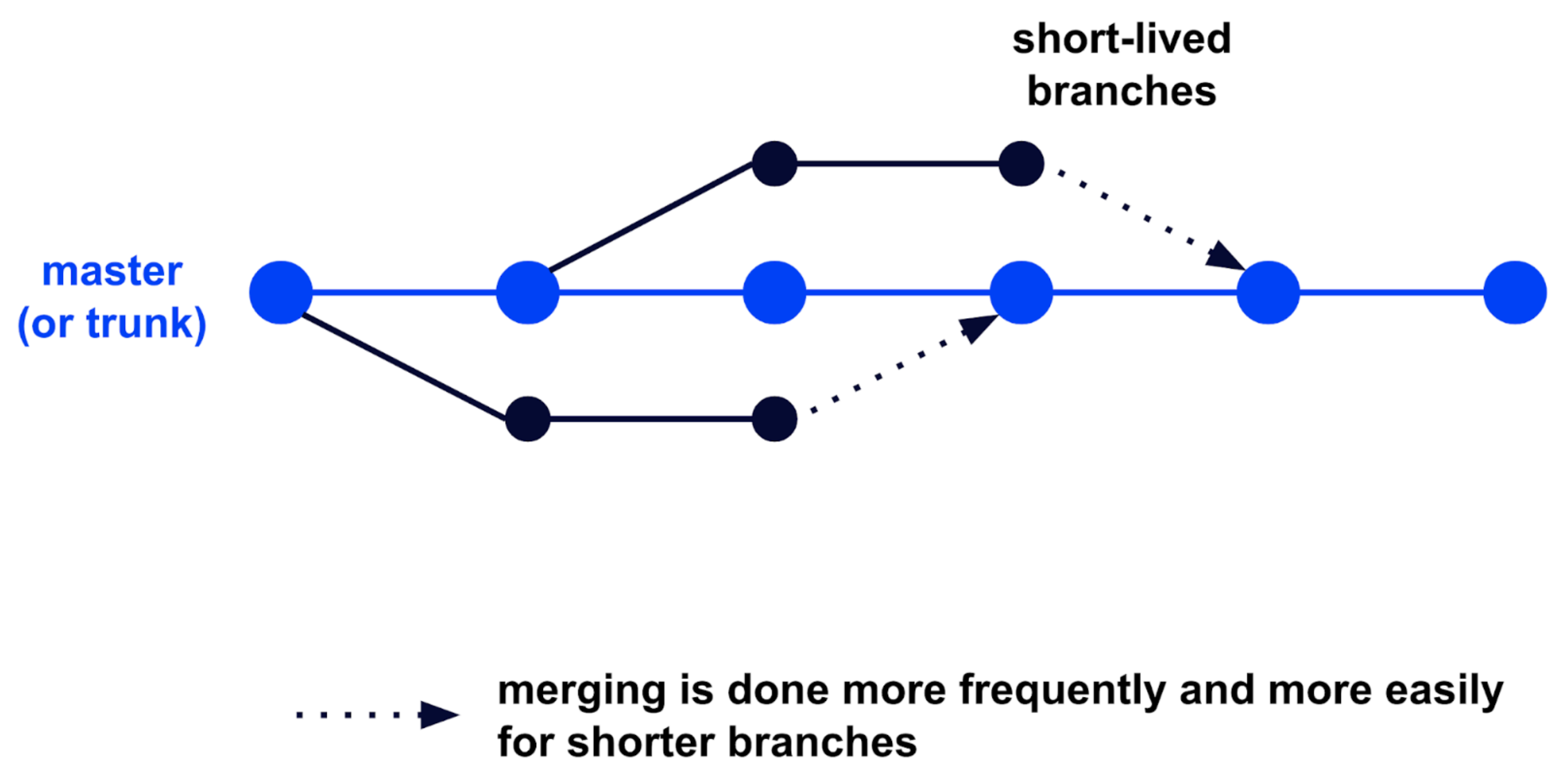


productboardbot on 30 Jul • edited

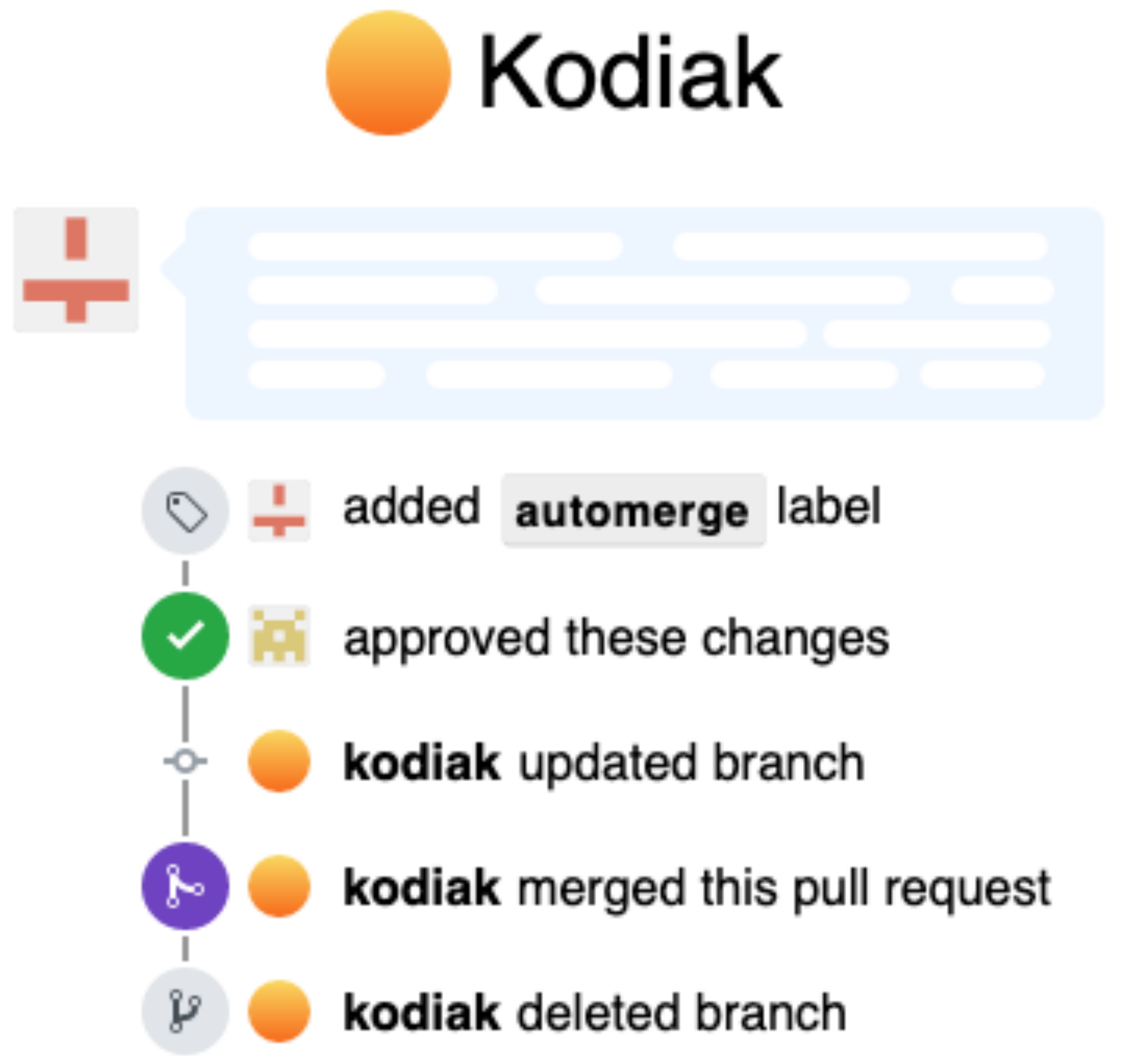
	Messages
📖	Link to affected dependency graph
📖	• Affected projects (2): • pb-home-data-access-product-ops • pb-home-feature-product-ops
📖	💚No dependency introduced

Generated by 🚫dangerJS against `0a980b6`

Kodiak


short-lived branches
master (or trunk)
merging is done more frequently and more easily for shorter branches

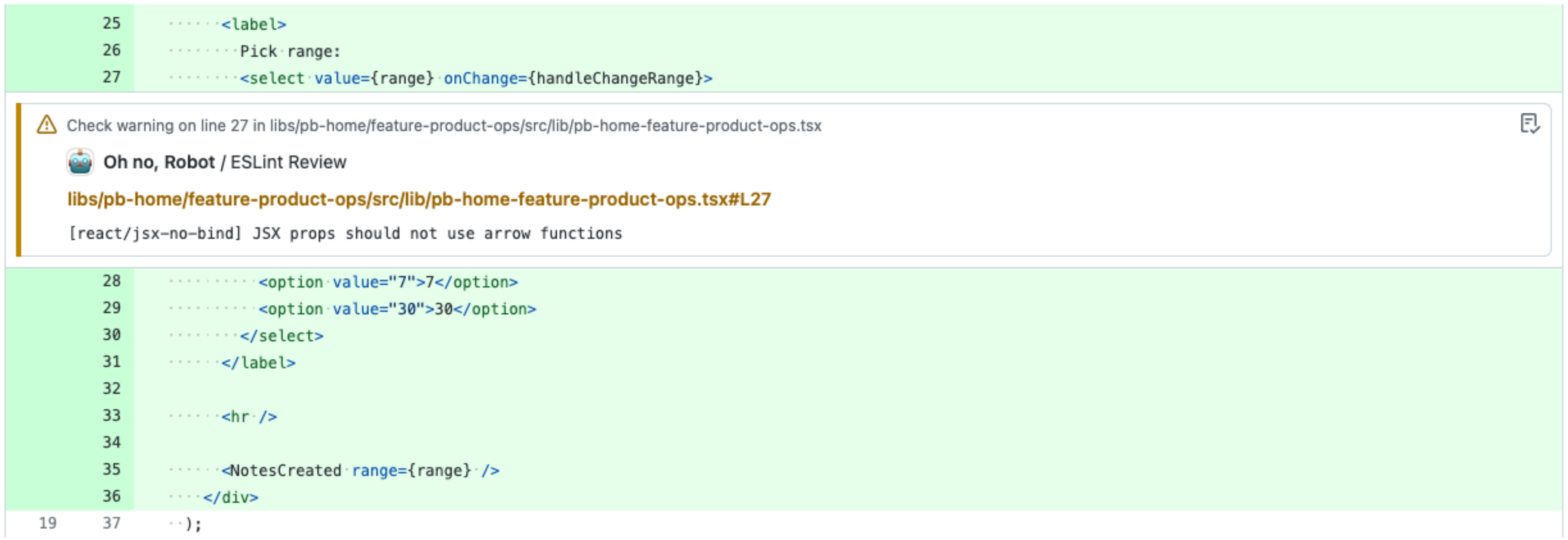
- Trunk based development
- Integration on feature branch
- Make robots do chores! 🤖



added automerge label
approved these changes
kodiak updated branch
kodiak merged this pull request
kodiak deleted branch

https://github.com/chdsbd/kodiak

Github Annotations



```
      <label>
        Pick range:
        <select value={range} onChange={handleChangeRange}>
```

Check warning on line 27 in libs/pb-home/feature-product-ops/src/lib/pb-home-feature-product-ops.tsx

Oh no, Robot / ESLint Review

libs/pb-home/feature-product-ops/src/lib/pb-home-feature-product-ops.tsx#L27

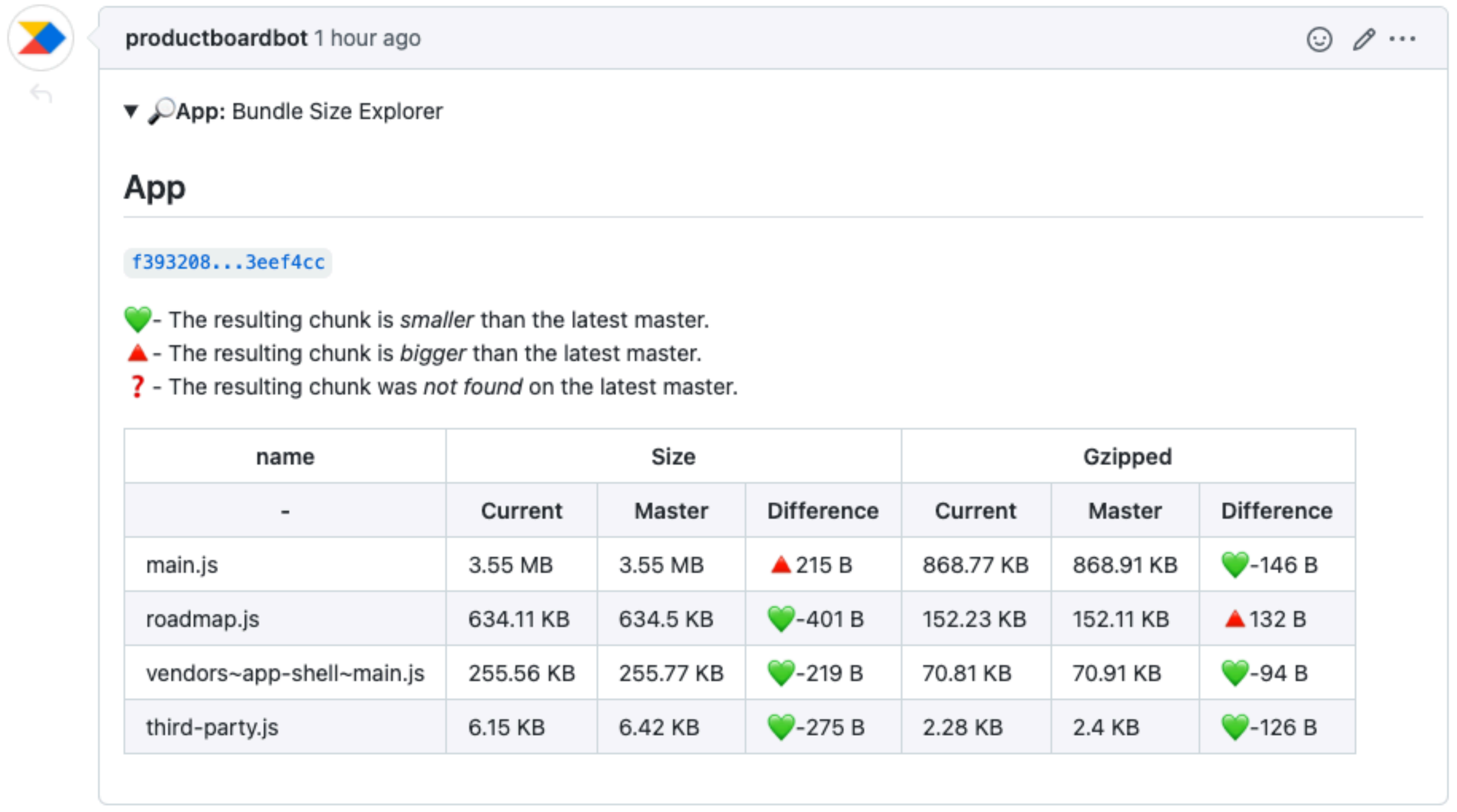
```
[react/jsx-no-bind] JSX props should not use arrow functions
```

```
          <option value="7">7</option>
          <option value="30">30</option>
        </select>
      </label>

      <hr />

      <NotesCreated range={range} />
    </div>
  );
```

Developers like to ignore CI outputs, bring it closer....

Bundle Size

- Stay on top of bundle size
- Bundle size budgets
- Warn you in case you bundle something huge



productboardbot 1 hour ago

▼ 🔎**App:** Bundle Size Explorer

App

`f393208...3eef4cc`

💚- The resulting chunk is *smaller* than the latest master.
🔺- The resulting chunk is *bigger* than the latest master.
❓ - The resulting chunk was *not found* on the latest master.

name	Size			Gzipped		
-	Current	Master	Difference	Current	Master	Difference
main.js	3.55 MB	3.55 MB	🔺215 B	868.77 KB	868.91 KB	💚-146 B
roadmap.js	634.11 KB	634.5 KB	💚-401 B	152.23 KB	152.11 KB	🔺132 B
vendors~app-shell~main.js	255.56 KB	255.77 KB	💚-219 B	70.81 KB	70.91 KB	💚-94 B
third-party.js	6.15 KB	6.42 KB	💚-275 B	2.28 KB	2.4 KB	💚-126 B

Takeaways

- Monorepos are not easy – at some scale you need dedicated team for it
- If the setup is right, it speeds up things – especially if your codebase is interconnected. Tooling has great impact. Nx proven to be great for our use case.
- You don't need to be FAANG to have a swag.

That's all Folks!

Thank you!

Q&A

- https://nx.dev
- Scaling monorepo — to infinity and beyond!
- How we measure adoption of a design system at Productboard
- https://danger.systems/js/
- https://github.com/chdsbd/kodiak

Jakub Beneš

Engineering Manager @ Productboard

@jukben

@jukben

https://jukben.codes