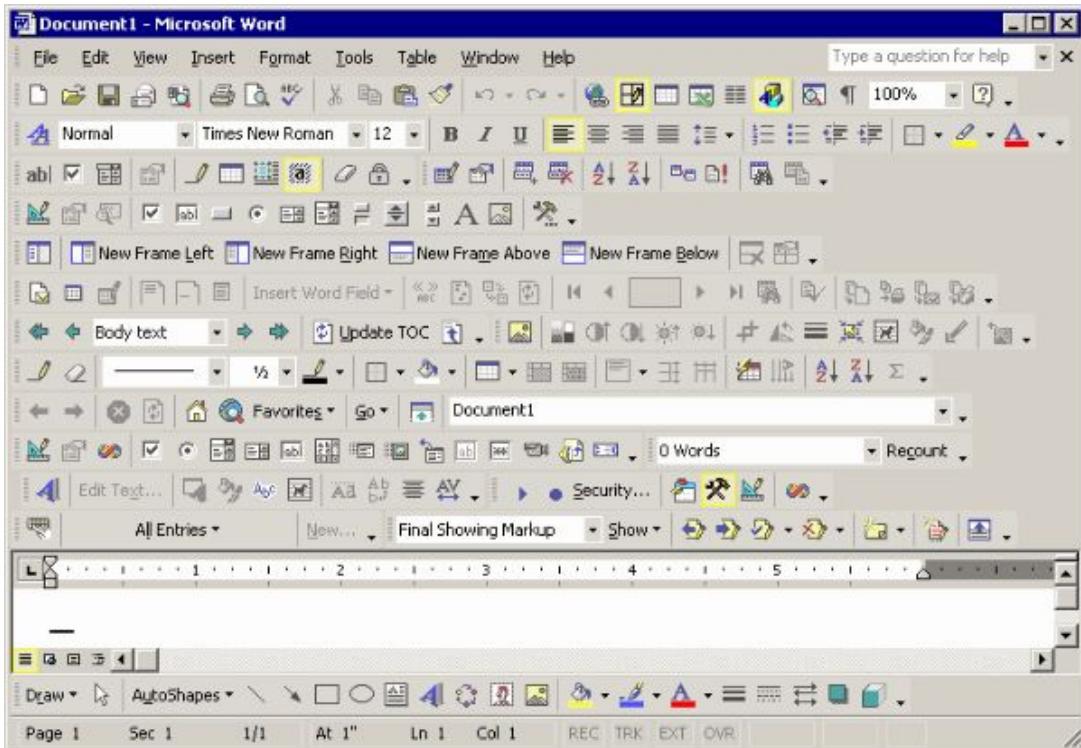


# **The unwritten rules for documenting, testing, and developing components with **Storybook****

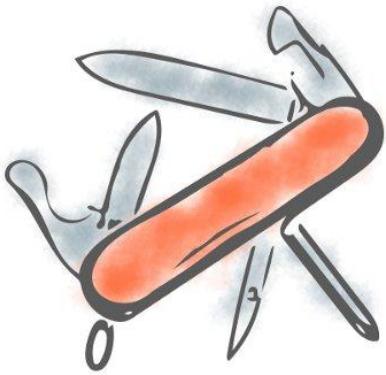


*Let's talk about*

# Feature Creep



*Microsoft word 2000*



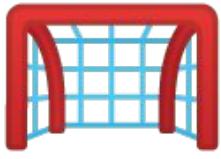
USERS  
NEEDS



STAKEHOLDERS  
IDEAS

<https://twitter.com/marvelapp/status/833871120495497218>





# Hello!

- Georgia Loper
- @developerLoper





Large enterprises



Tiny startups



Rapidly scaling companies



Large enterprises



Tiny startups



Rapidly scaling companies



Legacy systems



Static landing pages



SPAs



Web apps



Large enterprises



Tiny startups



Rapidly scaling companies



Legacy systems



Static landing pages



SPAs



Web apps

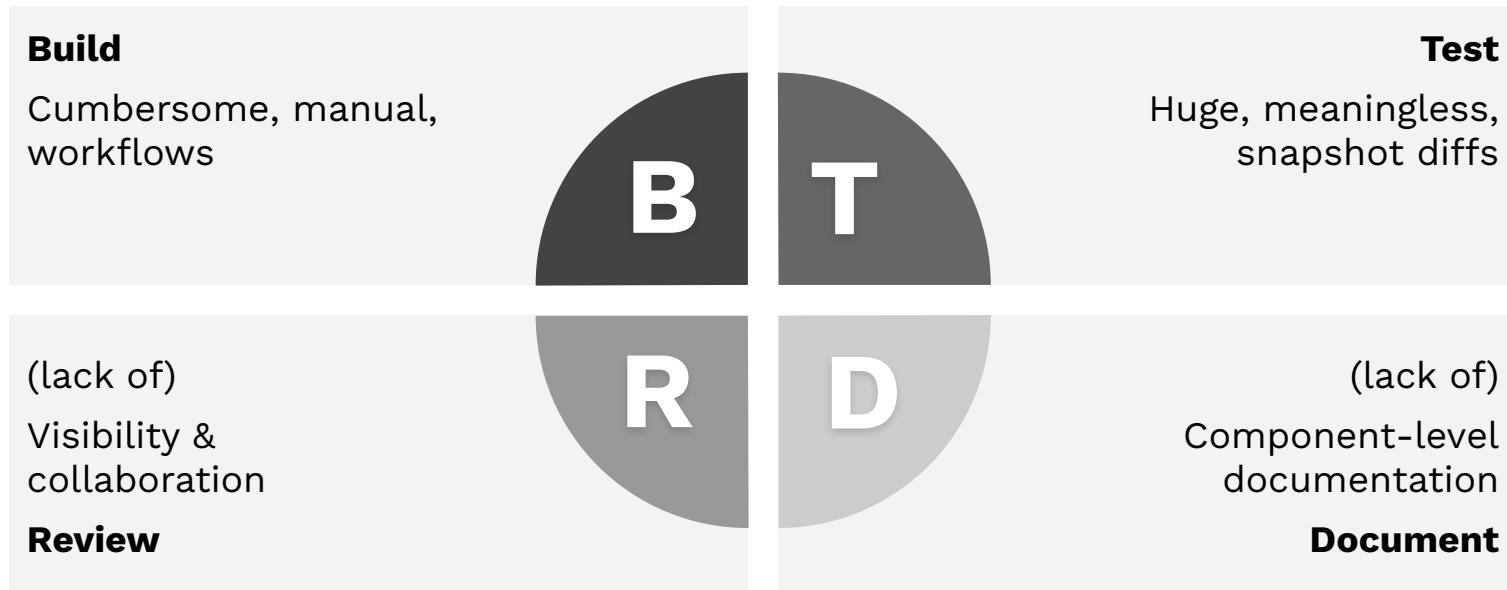


Patina Design System @ Upstart

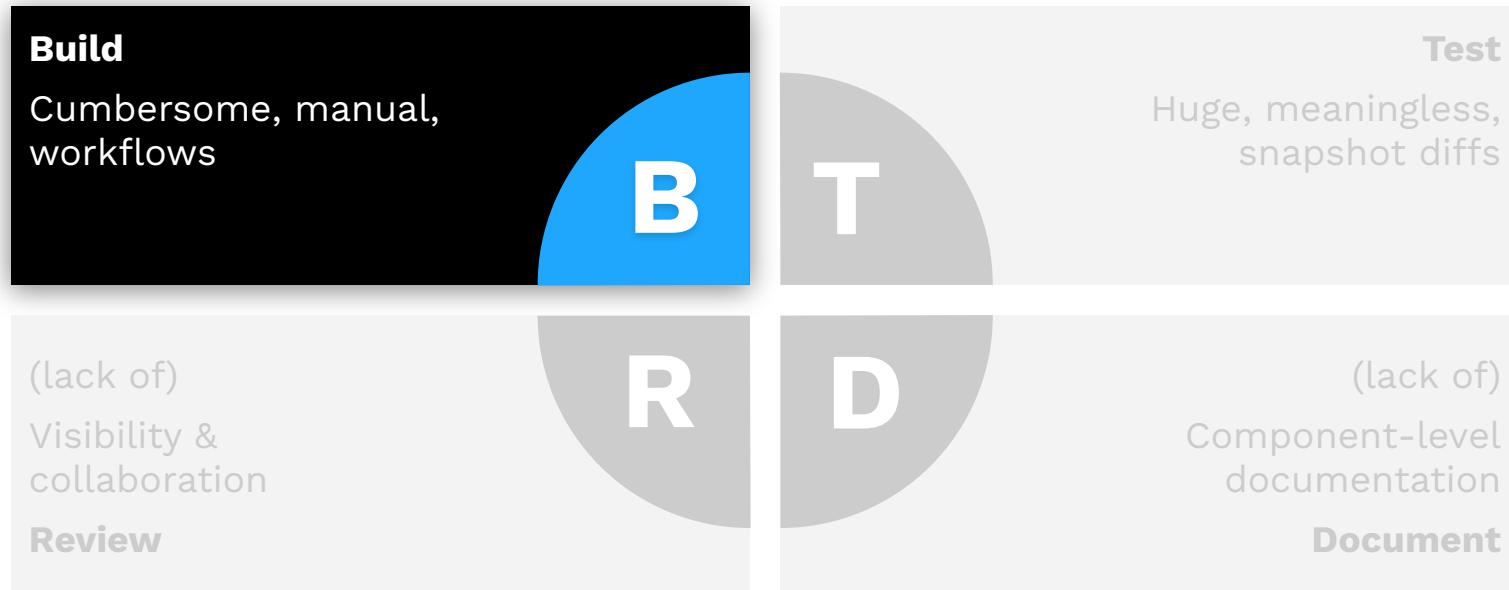
# Confession

I've released *a lot* of broken experiences

# The Software Development Life Cycle



# The Software Development Life Cycle

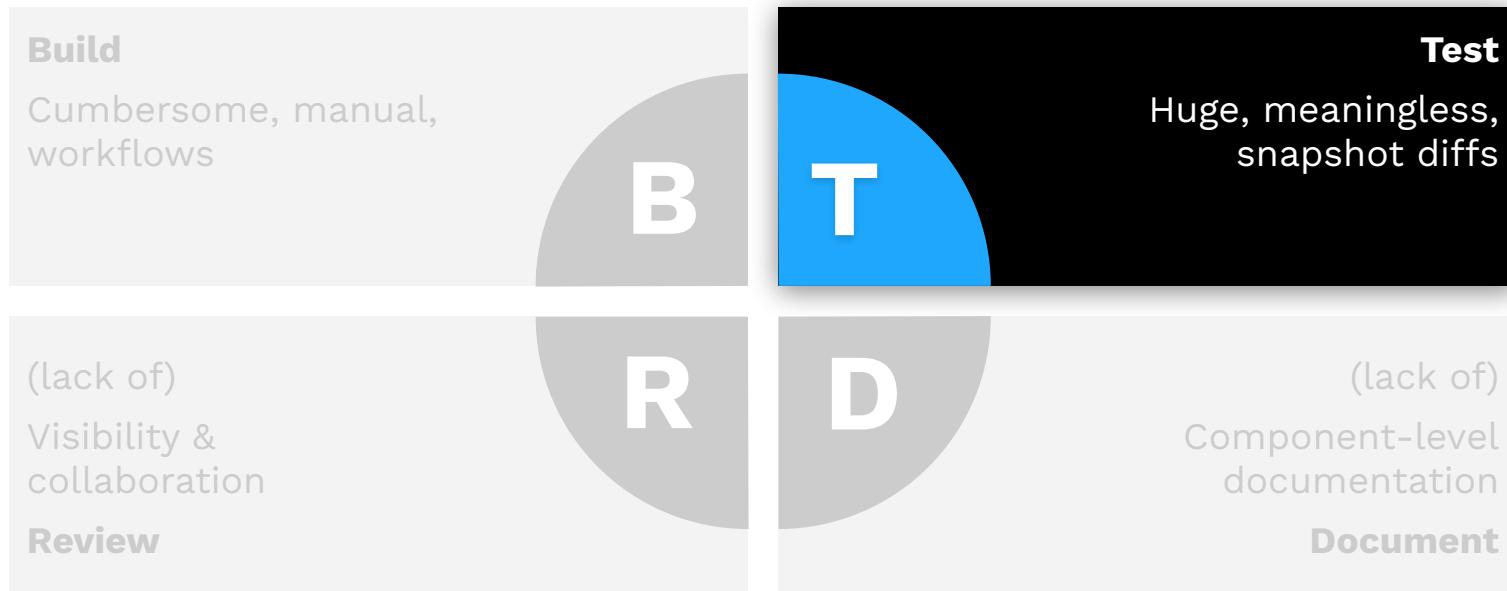


**we have to style**



**that edge case popup**

# The Software Development Life Cycle



The screenshot shows a terminal window with a dark theme. The title bar says "FAIL src/\_tests\_/Link.react-test.js". The main area displays the Jest test code and its snapshot diff:

```
• renders correctly

  expect(received).toMatchSnapshot()

  Snapshot name: `renders correctly 1`

  - Snapshot - 2
  + Received + 2

  <a
    className="normal"
  - href="http://www.facebook.com"
  + href="http://www.instagram.com"
    onMouseEnter={[Function]}
    onMouseLeave={[Function]}
  >
  - Facebook
  + Instagram
</a>
```

<https://jestjs.io/>

```
FAIL  src/components/TableList/__tests__/TableList.test.js
  ● <TableList /> > shows the item price if a price is present

    expect(value).toMatchSnapshot()

      Received value does not match stored snapshot 1.

      - Snapshot
      + Received

@@ -142,28 +142,24 @@
      >     className="emotion-8 emotion-9"
      >       Chicken Parm. Sandwhich
      </div>
      <div
      -   className="ui basic segment"
      +   className="emotion-10 emotion-9"
      >
      -   <div
      -     className="emotion-10 emotion-9"
      -   >
      -     <input
      -       style={
      -         Object {
      -           "width": "40px",
      -         }
      +     <input
      +       style={
      +         Object {
      +           "width": "40px",
      +         }
      -       type="text"
      -       value={2}
      -     />
      -   </div>
      -   <div
      -     className="emotion-10 emotion-9"
      -   >
      -     $11.99
      -   </div>
      +     <input
      +       type="text"
      +       value={2}
      +     />
      </div>
      <div
      +   className="emotion-10 emotion-9"
      +   >
      +     $11.99
      +   </div>
      </div>

      at Object.<anonymous> (src/components/TableList/__tests__/TableList.test.js:43:23)
        at new Promise (<anonymous>)
        at Promise.resolve.then.el (node_modules/p-map/index.js:46:16)
          at <anonymous>
          at process._tickDomainCallback (internal/process/next_tick.js:228:7)

<TableList />
  ✕ shows the item price if a price is present (8ms)
```

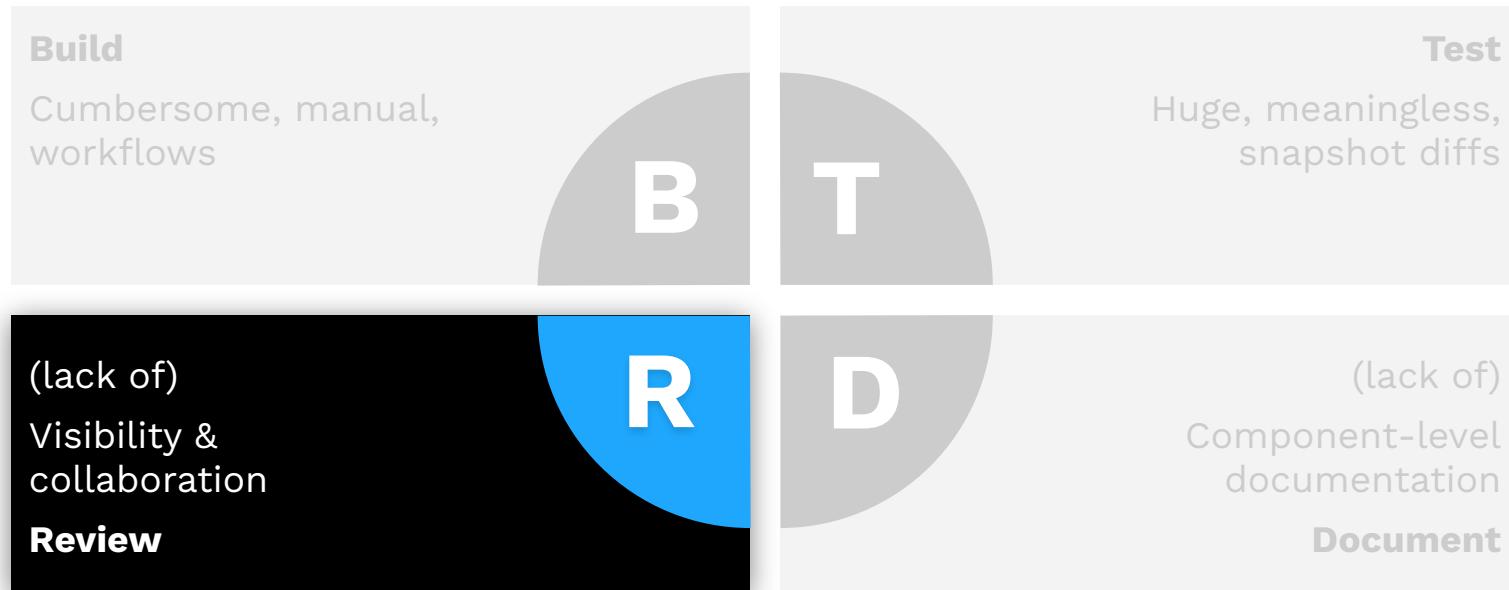
#### Snapshot Summary

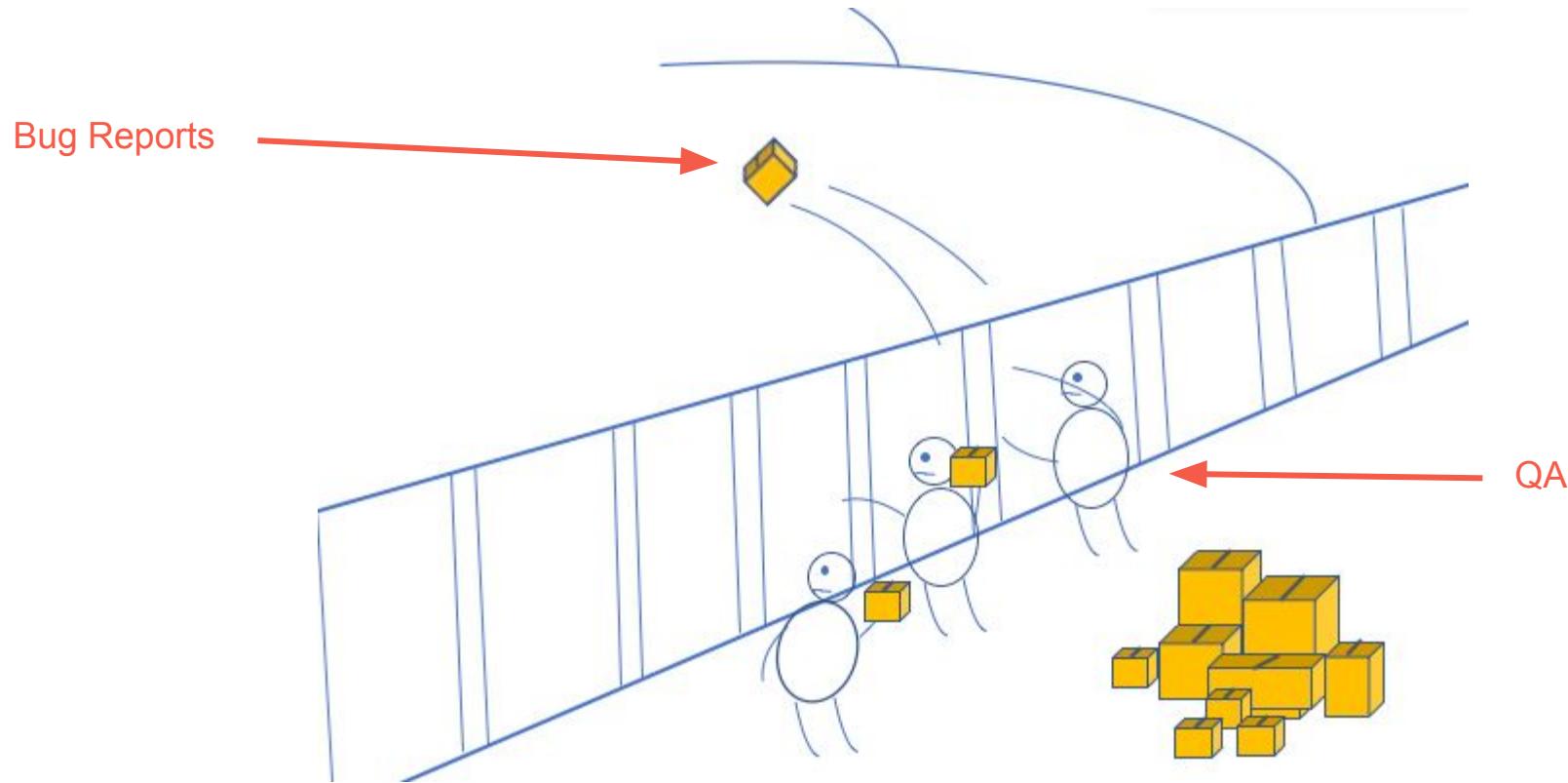
> 1 snapshot test failed in 1 test suite. Inspect your code changes or press 'u' to update them.

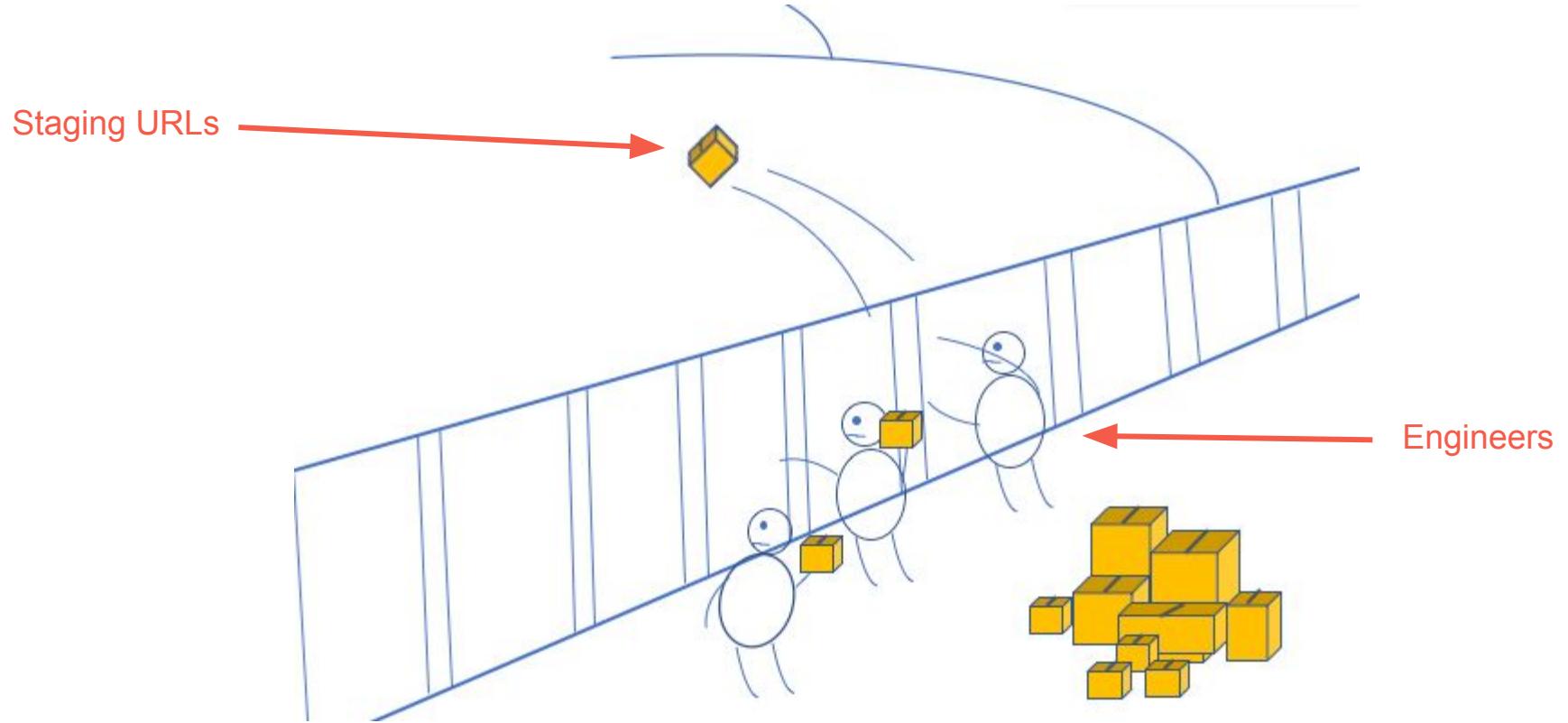
**YOU GET A LGTM! YOU GET A LGTM!**

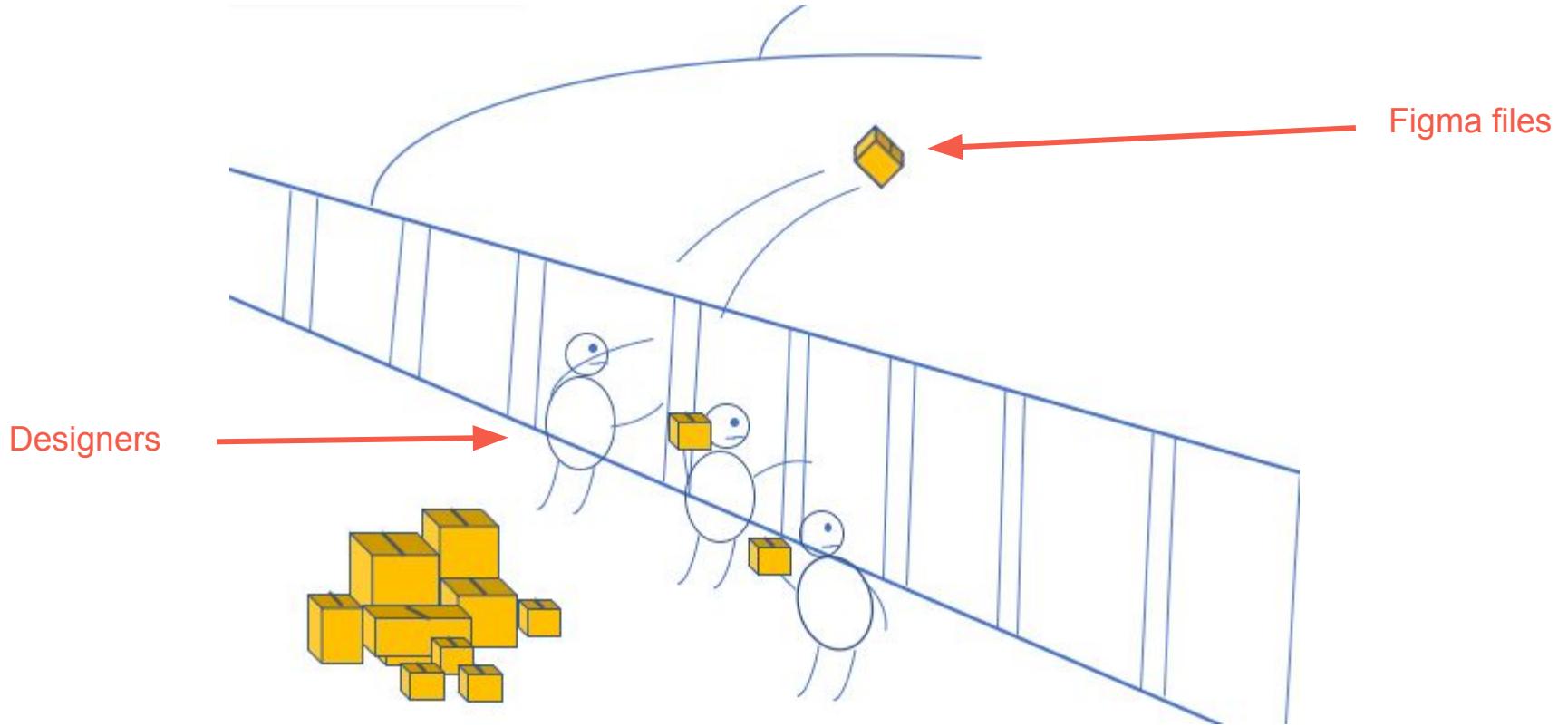
**AND YOU GET A LGTM!**

# The Software Development Life Cycle









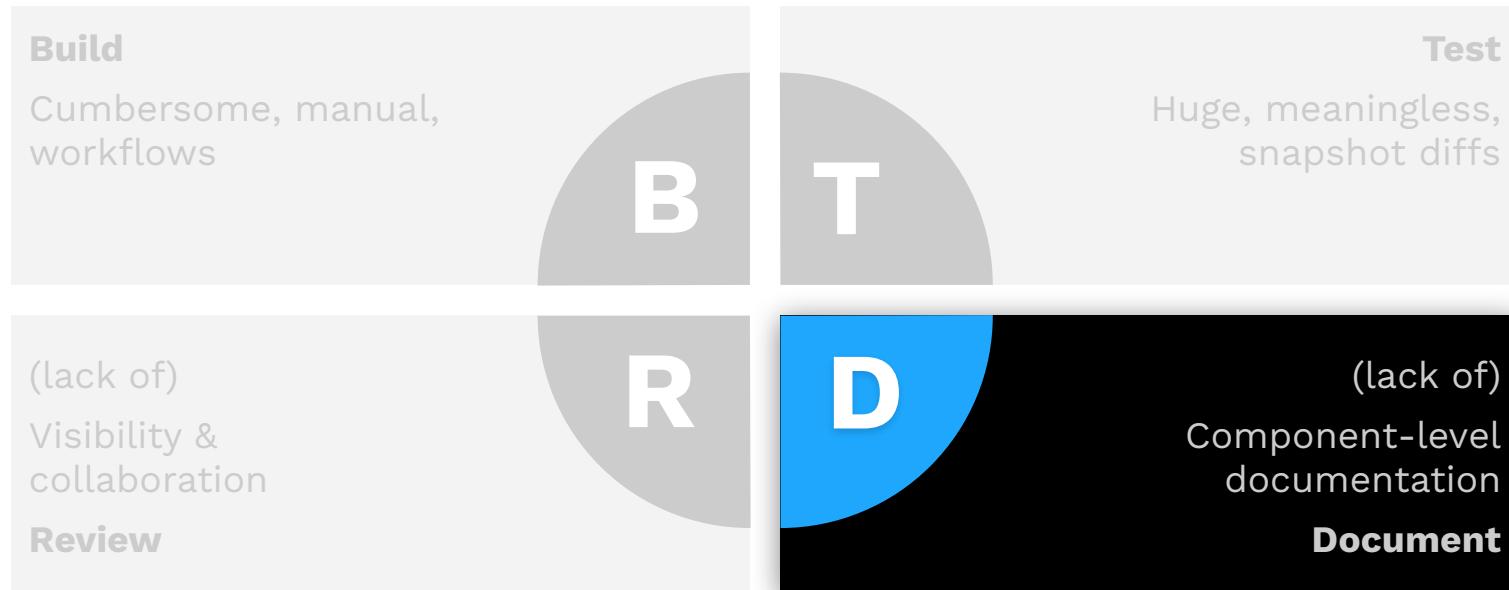


An error occurred.

NO

OK

# The Software Development Life Cycle



Client: Asks for a special case logic to remove some data when it matches some edge case

Also client: Files a bug three months later when they see data removed

Dev: Tells client data matches edge case, he requested the very same behaviour

Client:



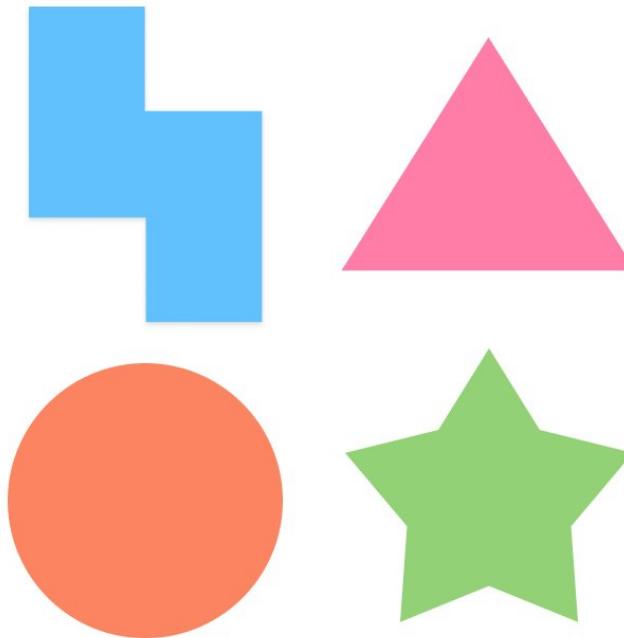
Working as intended

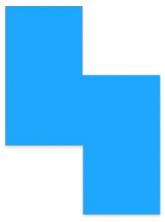
**How do we do  
better?**

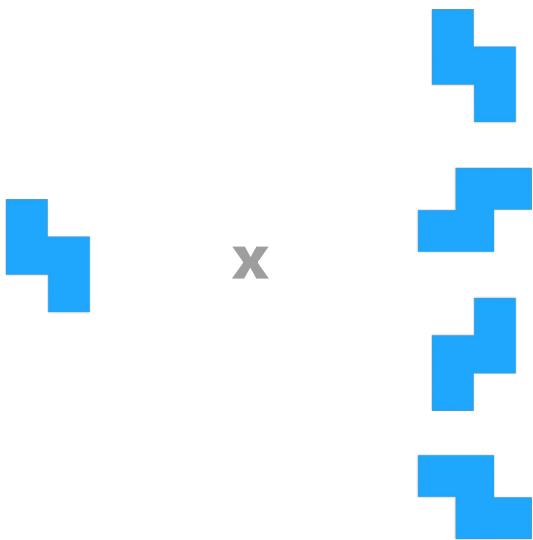
*Build*

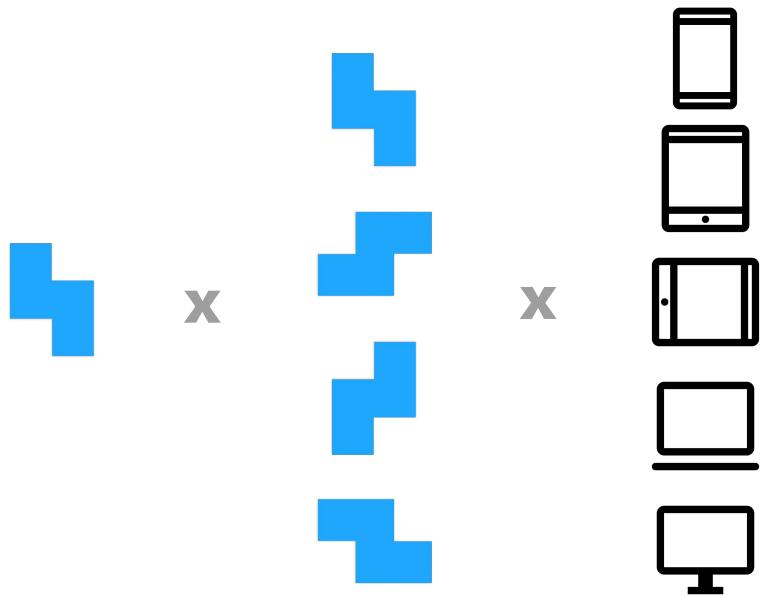
# Storybook as a **workshop** environment

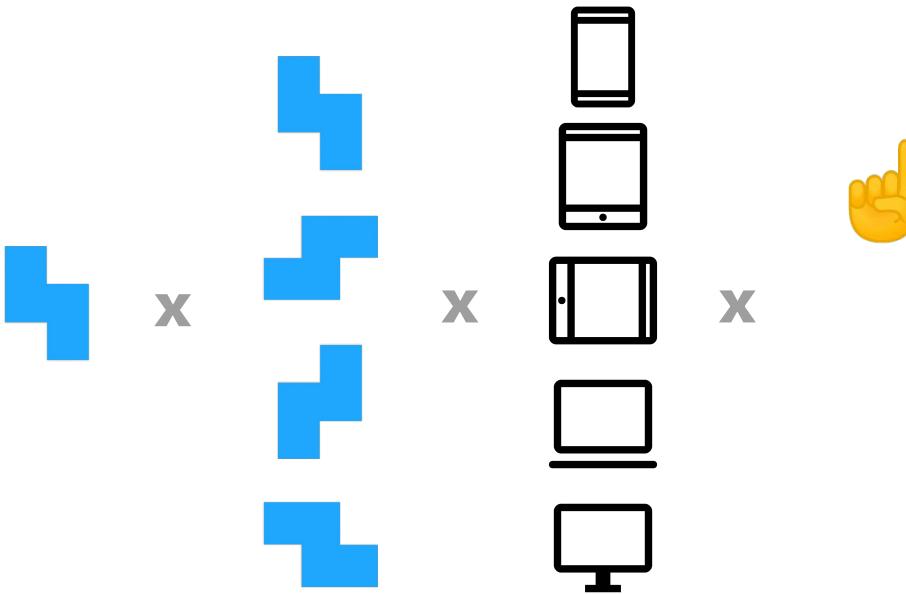
**Let's talk  
about  
components**

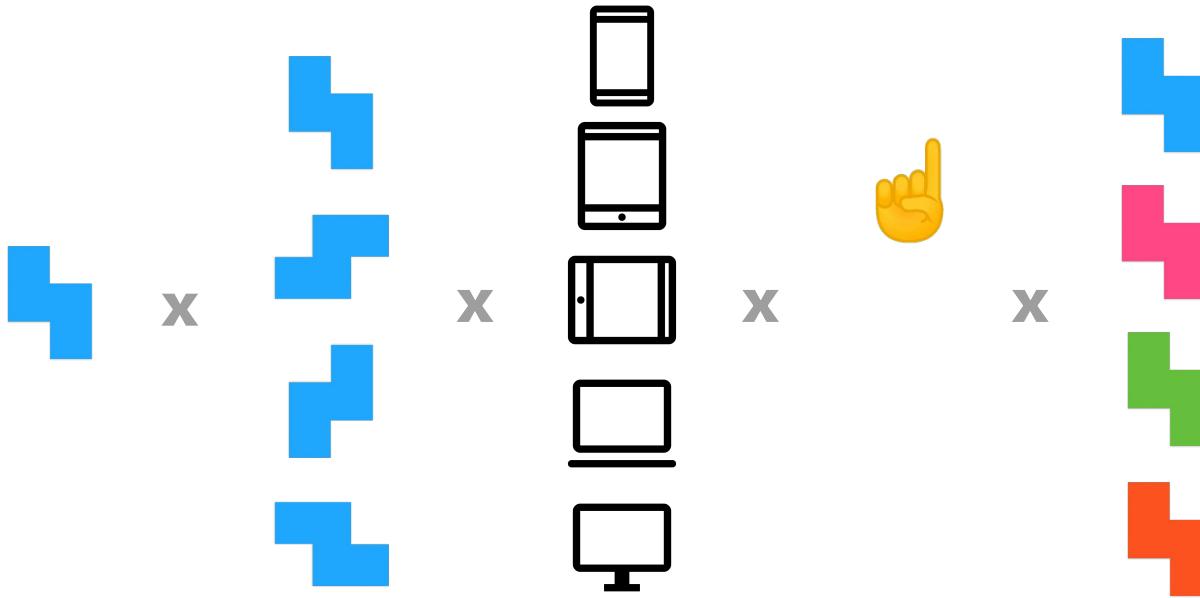


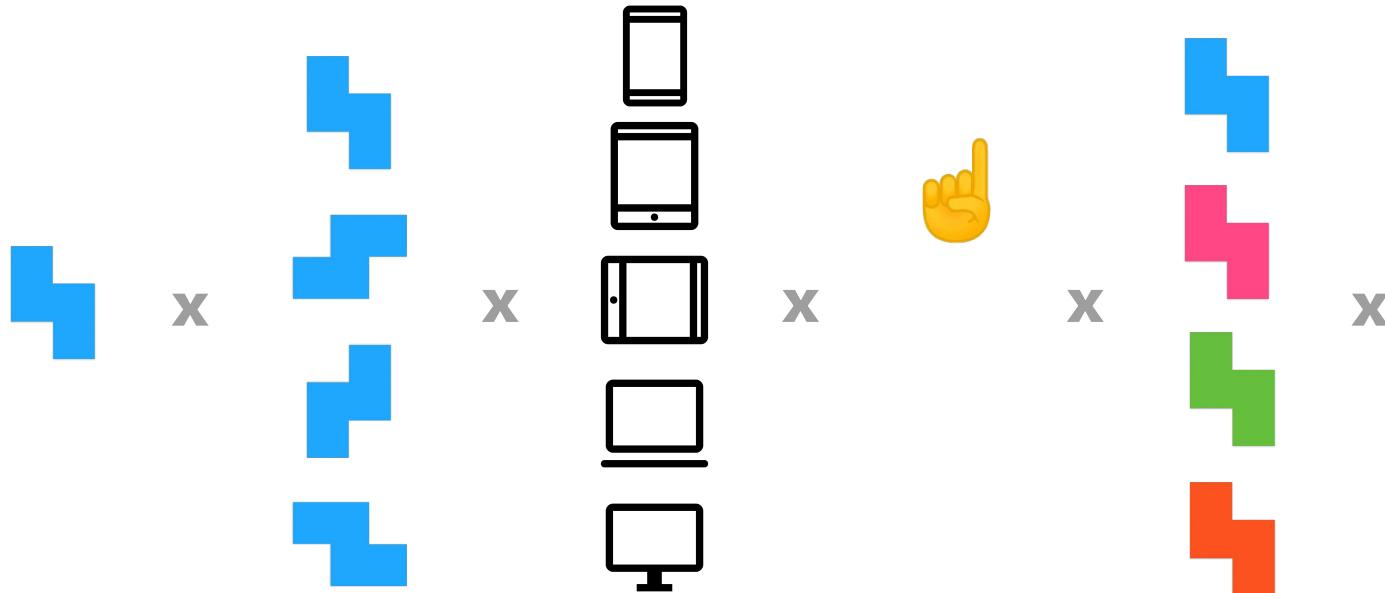


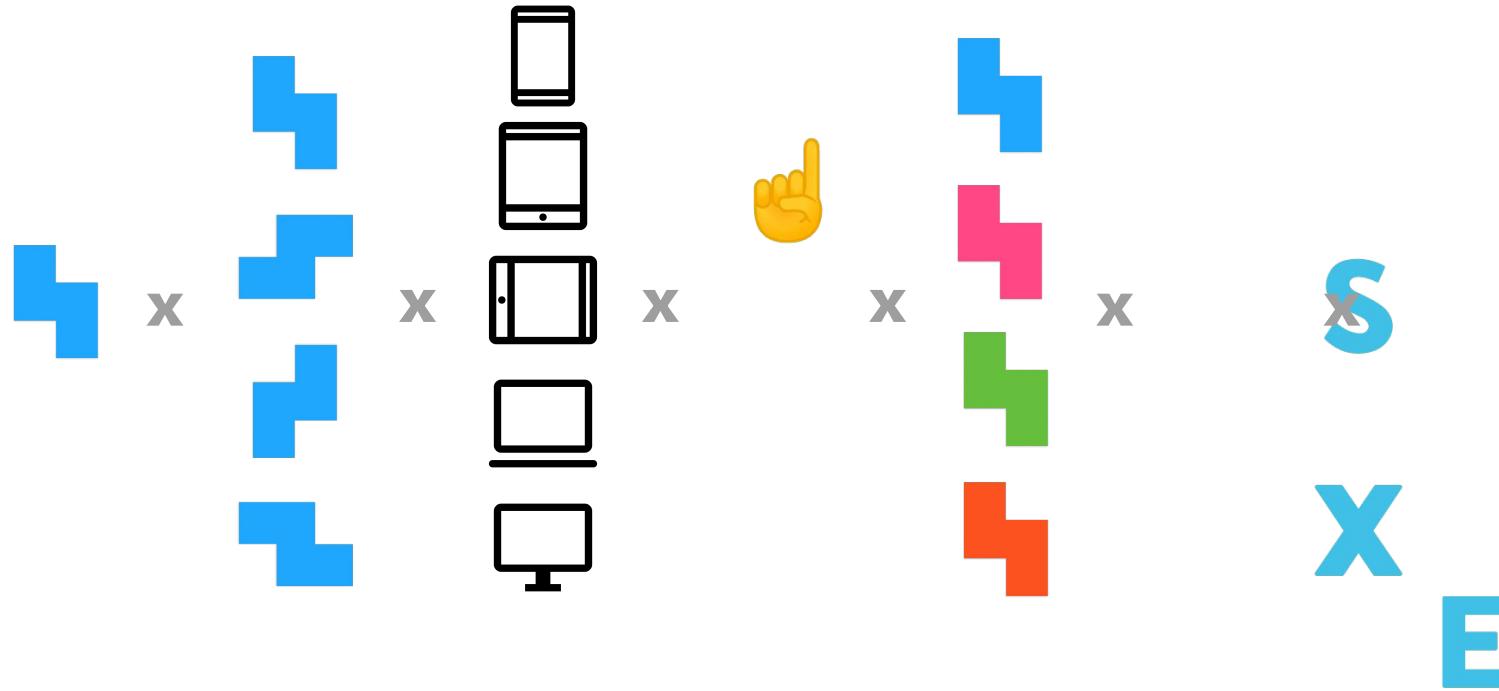


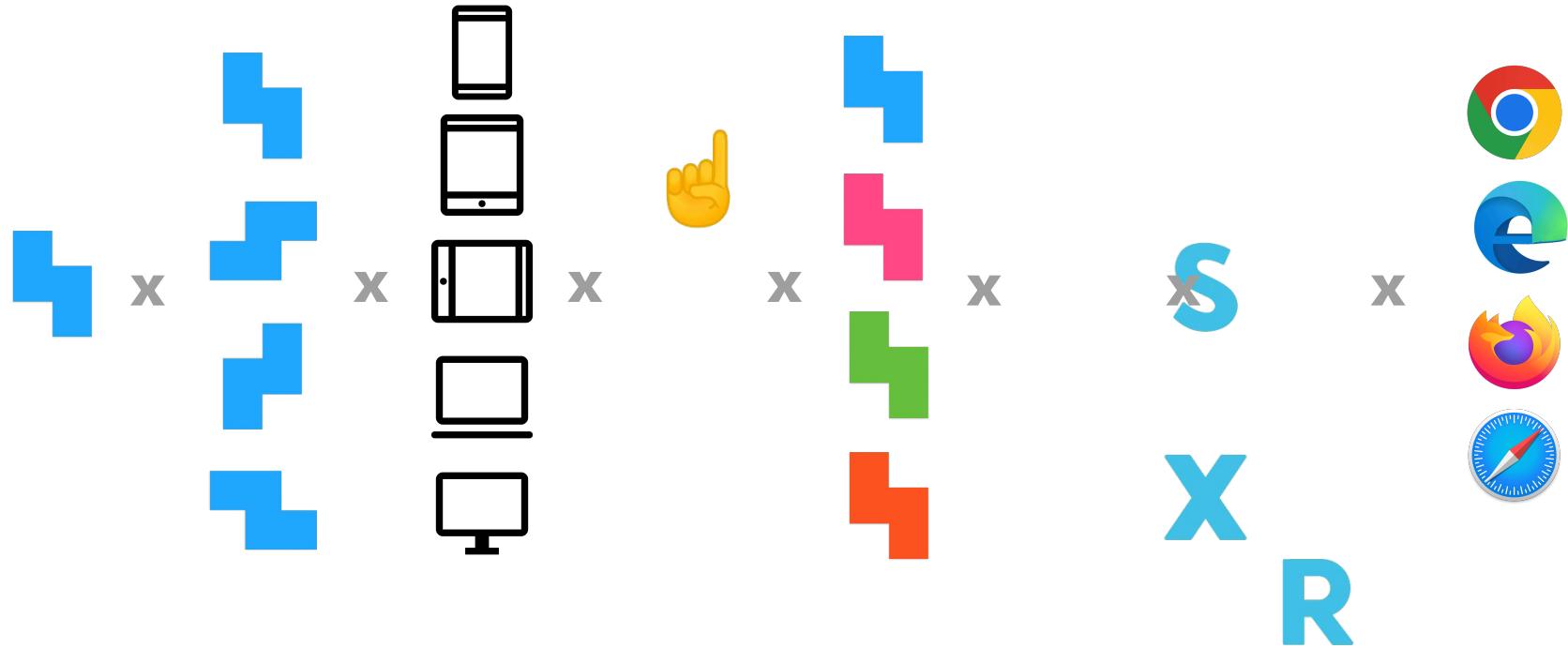




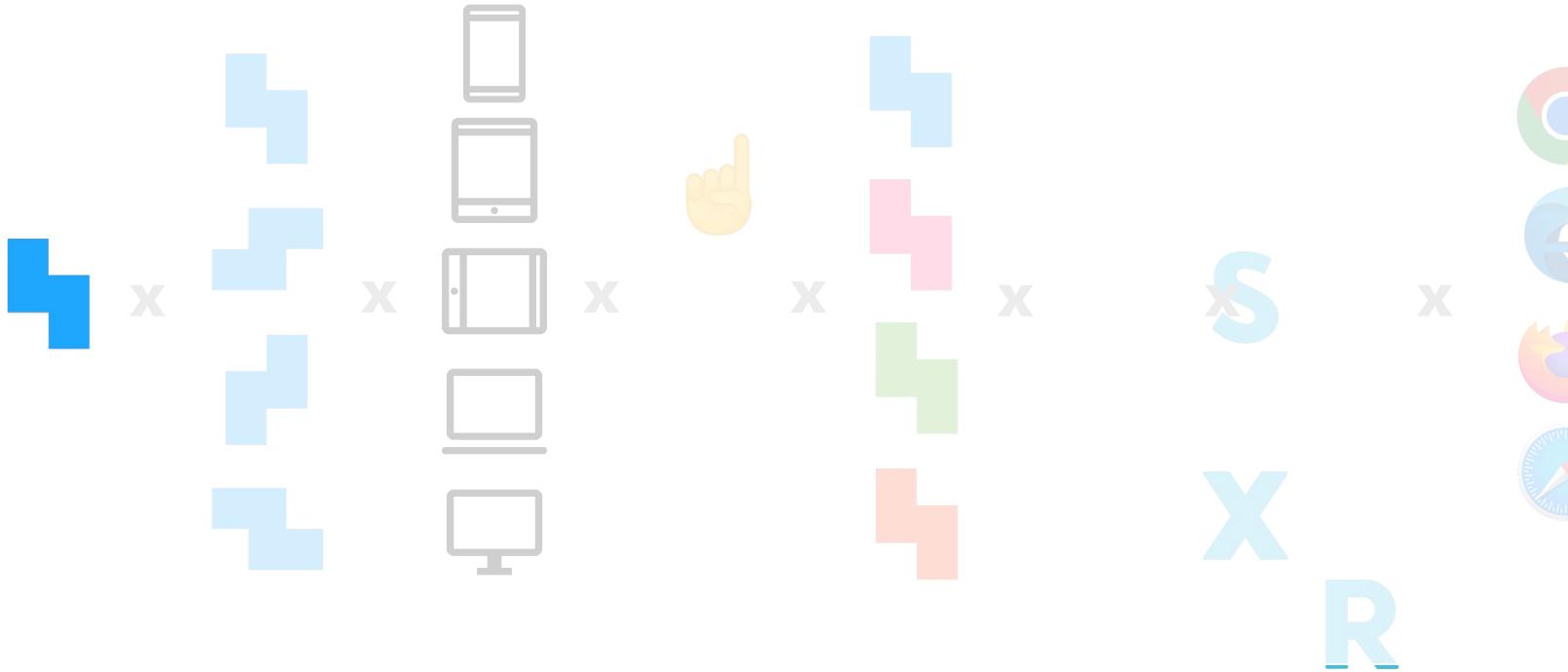














```
// Button.stories.jsx
import Button from './Button';

export default {
  title: "Example/Button",
  component: Button
}
```



# Storybook

...



Find components

/

## EXAMPLE



Introduction

Button

Header

Page

Canvas

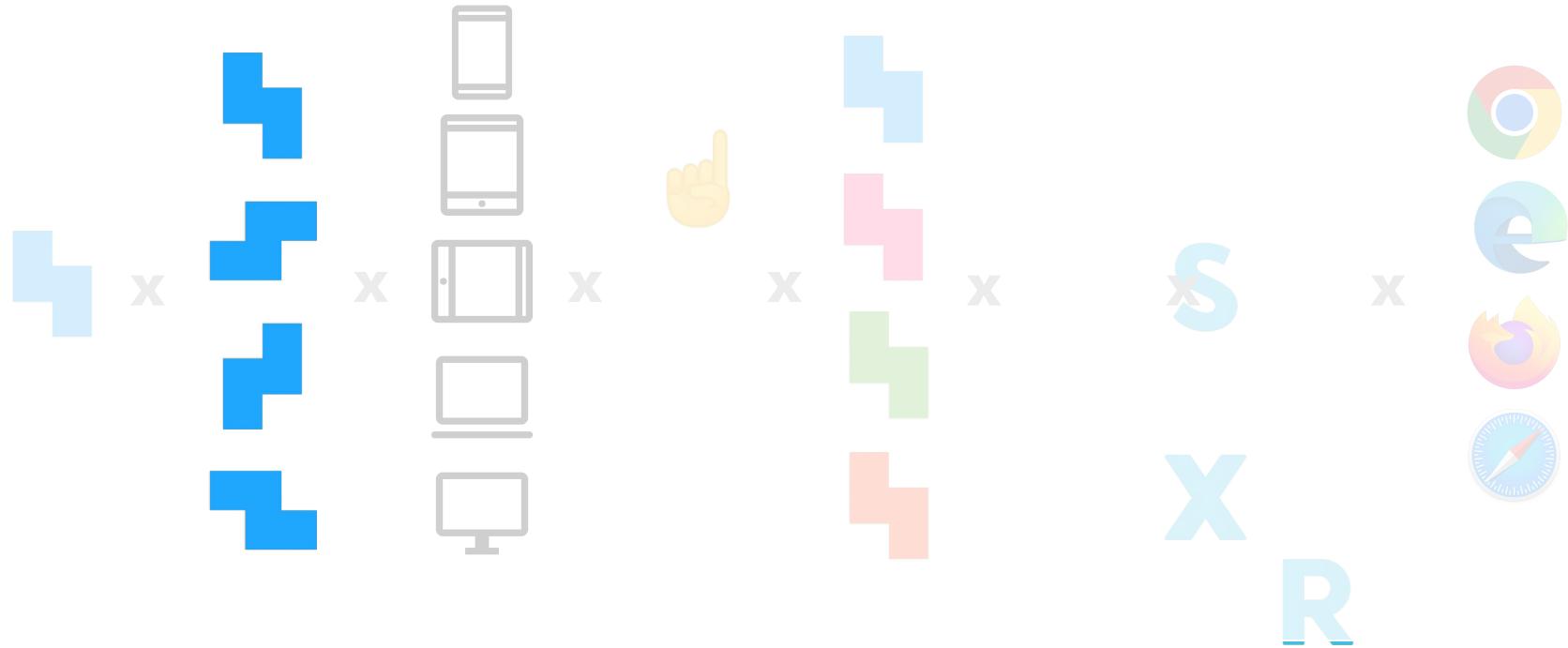
Docs

Button



```
// Button.stories.jsx
import Button from './Button';

export default {
  title: "Example/Button",
  component: Button
}
```





```
// Button.stories.jsx
import React from 'react'

export default { ... }

// CSF 2
const Template = (args) => <Button {...args}>

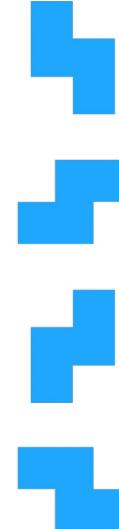
export const Primary = Template.bind({})
Primary.args = {
  primary: true,
  label: "Button"
}
```

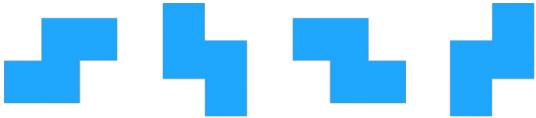


```
// Button.stories.jsx
import Button from './Button'

export default { ... }

// CSF 3
export const Primary = {
  args: {
    primary: true,
    label: "Button"
  }
}
```





Mac OS X style window title bar with red, yellow, and green buttons.

**S Storybook** ...

Find components /

**EXAMPLE** ⌄

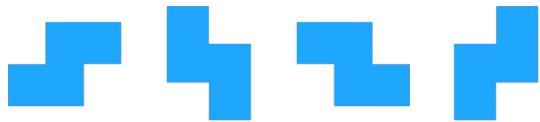
- Introduction
- Button**
- Primary**
- Secondary
- Large
- Small

Header

Page

The screenshot shows the Storybook application interface. The top navigation bar includes tabs for "Canvas" (which is active) and "Docs", along with various icons for search, refresh, and layout. The main content area is titled "Button". On the left, a sidebar titled "EXAMPLE" lists components: "Introduction", "Button" (selected and highlighted in blue), "Primary" (also selected and highlighted in blue), "Secondary", "Large", and "Small". Below this, there are collapsed sections for "Header" and "Page". A search bar at the top left contains the placeholder "Find components" and a slash character. The overall design is clean and modern, with a focus on component-based development.

The screenshot shows the Storybook interface. The top navigation bar includes tabs for 'Canvas' (which is active) and 'Docs'. Below the navigation is a search bar labeled 'Find components' with a placeholder 'Search' and a clear button. On the left, a sidebar titled 'EXAMPLE' lists various component categories: 'Introduction', 'Button' (selected), 'Primary', 'Secondary', 'Large' (selected), 'Small', 'Header', 'Logged In', 'Logged Out', 'Page', 'Logged Out', and 'Logged In'. The main content area displays a large button component with the label 'Button'. Below the component is a control panel titled 'Controls (4)' with tabs for 'Actions' and 'Interactions'. The 'size' control is set to 'large' (radio button selected). The 'label\*' control shows the value 'Button'. The 'primary' control has a 'Set boolean' button. The 'backgroundColor' control shows a color picker with the placeholder 'Choose color...'. The 'onClick' control has a '-' button.



Storybook

Canvas Docs

Find components /

EXAMPLE

- Introduction
- Button
  - Primary
  - Secondary
  - Large
  - Small
- Header
- Page

Enable measure

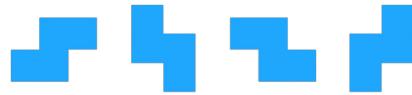
16

Button

561 x 36

16

16



Storybook

Canvas Docs

Find components /

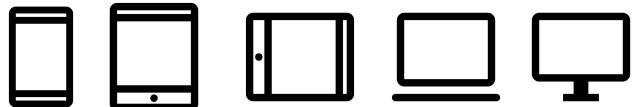
Acme

Welcome, Jane Doe! Log out

EXAMPLE

- Introduction
- Button
  - Primary
  - Secondary
  - Large
  - Small
- Header
  - Logged In
  - Logged Out
- Page





Storybook

Canvas Docs

Large mobile (P) 414 896

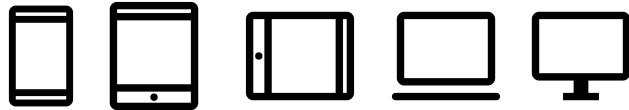
Find components /

EXAMPLE

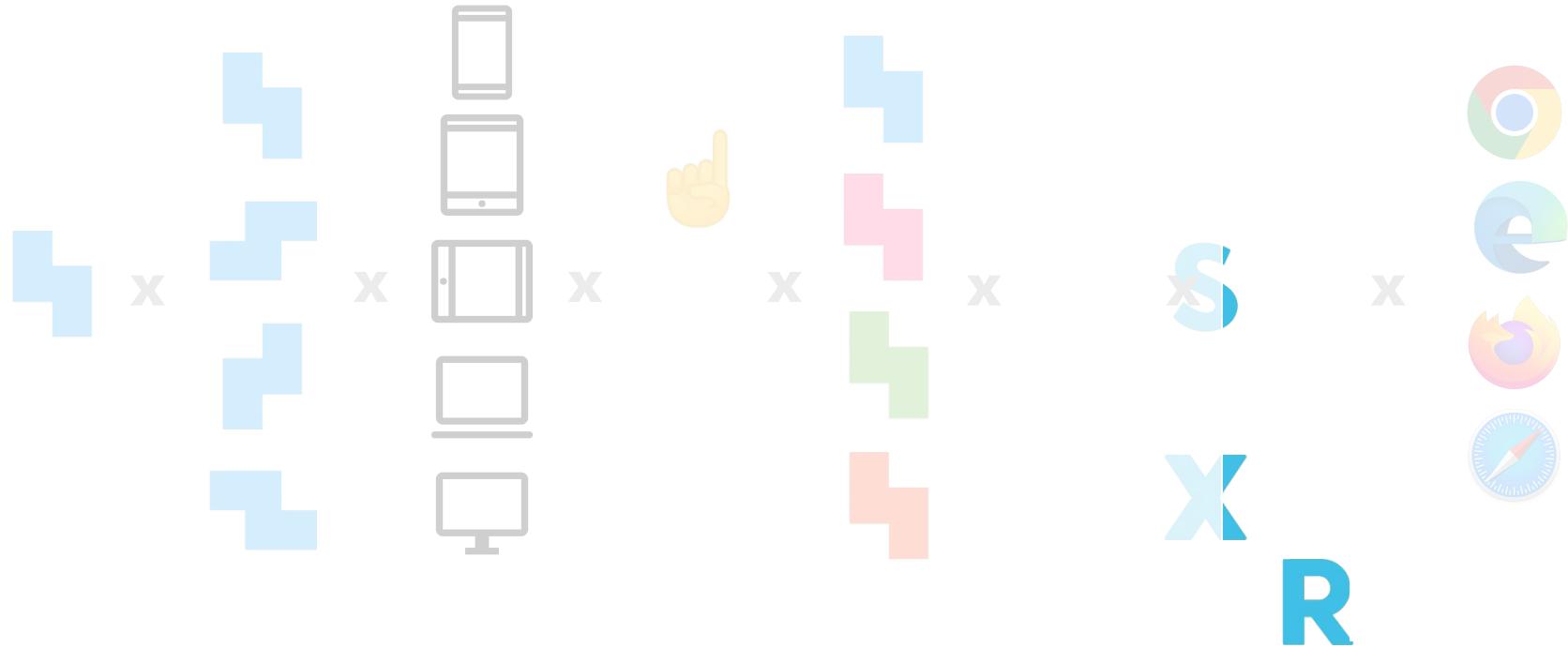
- Introduction
- Button
  - Primary
  - Secondary
  - Large
  - Small
- Header
- Logged In
  - Logged Out
- Page

Acme Welcome, Jane Doe! Log out

The screenshot shows the Storybook interface. On the left, there's a sidebar with a tree view of components under 'EXAMPLE'. The 'Logged In' section is currently selected, showing two sub-options: 'Logged Out' (disabled) and 'Page'. The main area is a 'Canvas' where a component from the 'Logged In' section is being previewed. The preview shows a header with the 'Acme' logo and the text 'Welcome, Jane Doe!' followed by a 'Log out' button. The preview is set to 'Large mobile (P)' and has dimensions of 414x896. At the top, there are standard browser-like controls for refresh, search, and zoom.



The screenshot shows the Storybook application interface. The top navigation bar includes tabs for 'Canvas' (which is selected), 'Docs', and 'Tablet (P)'. There are also icons for search, refresh, and other settings. The left sidebar, titled 'EXAMPLES', lists components: 'Introduction', 'Button' (with sub-options 'Primary', 'Secondary', 'Large', 'Small'), 'Header' (with sub-options 'Logged In' which is selected, and 'Logged Out'), and 'Page'. The main canvas area displays a header component for 'Acme' with a logo, the text 'Acme', a welcome message 'Welcome, Jane Doe!', and a 'Log out' button.





Storybook

Find components /

EXAMPLE

- Introduction
- Button
  - Primary
  - Secondary
  - Large
  - Small
- Header
- Page

Canvas Docs

Change the background of the preview

Button

A screenshot of the Storybook application interface. The left sidebar shows a navigation tree with sections like 'EXAMPLE', 'Introduction', 'Button' (which is expanded), 'Header', and 'Page'. Under 'Button', 'Primary' is selected and highlighted in blue. The main canvas area displays a single blue button component with the word 'Button' on it. A tooltip 'Change the background of the preview' is visible above the canvas. The top bar includes tabs for 'Canvas' and 'Docs', as well as various tool icons.





```
// .storybook/preview.js
export const parameters = {
  actions: { argTypesRegex: "^on[A-Z].*$" },
  controls: {
    matchers: {
      color: /(background|color)$/i,
      date: /Date$/,
    },
  },
}
```



```
// .storybook/preview.js
export const parameters = {}

export const globalTypes = {
  locale: {
    name: 'Locale',
    defaultValue: 'en',
    toolbar: {
      icon: 'globe',
      dynamicTitle: true,
      items: [
        { value: 'en', title: 'English' },
        { value: 'es', title: 'Spanish' },
      ],
    },
  },
}
```



```
● ● ●  
  
// .storybook/preview.js  
import { LocaleProvider } from '*/LocaleProvider'  
  
export const parameters = {}  
  
export const globalTypes = {}  
  
export const decorators = [  
  (Story, context) => {  
    const { locale } = context.globals  
    return (  
      <LocaleProvider locale={locale}>  
        <Story />  
      </LocaleProvider>  
    )  
  },  
]
```



Storybook

Canvas Docs

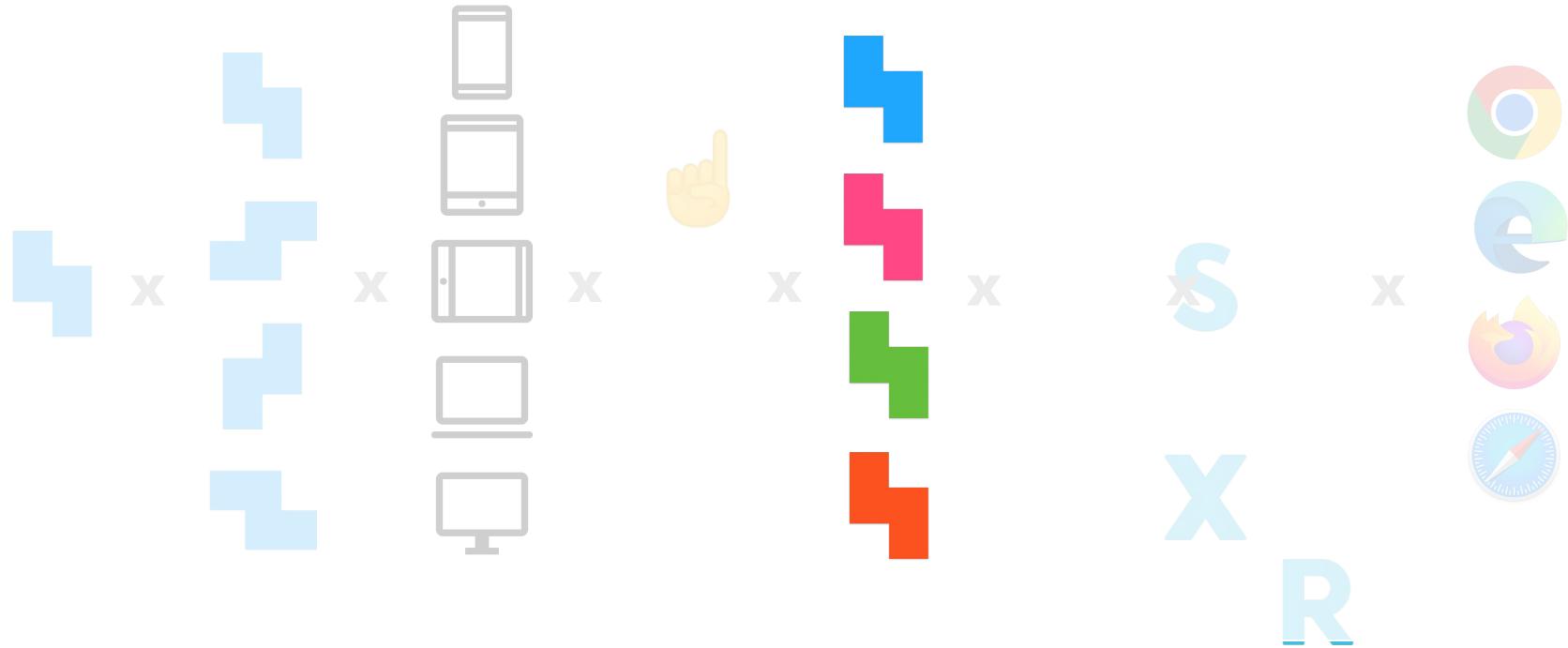
Find components /

EXAMPLE

- Introduction
- Button
  - Primary
  - Secondary
  - Large
  - Small
- Header
- Page

Botón

Spanish



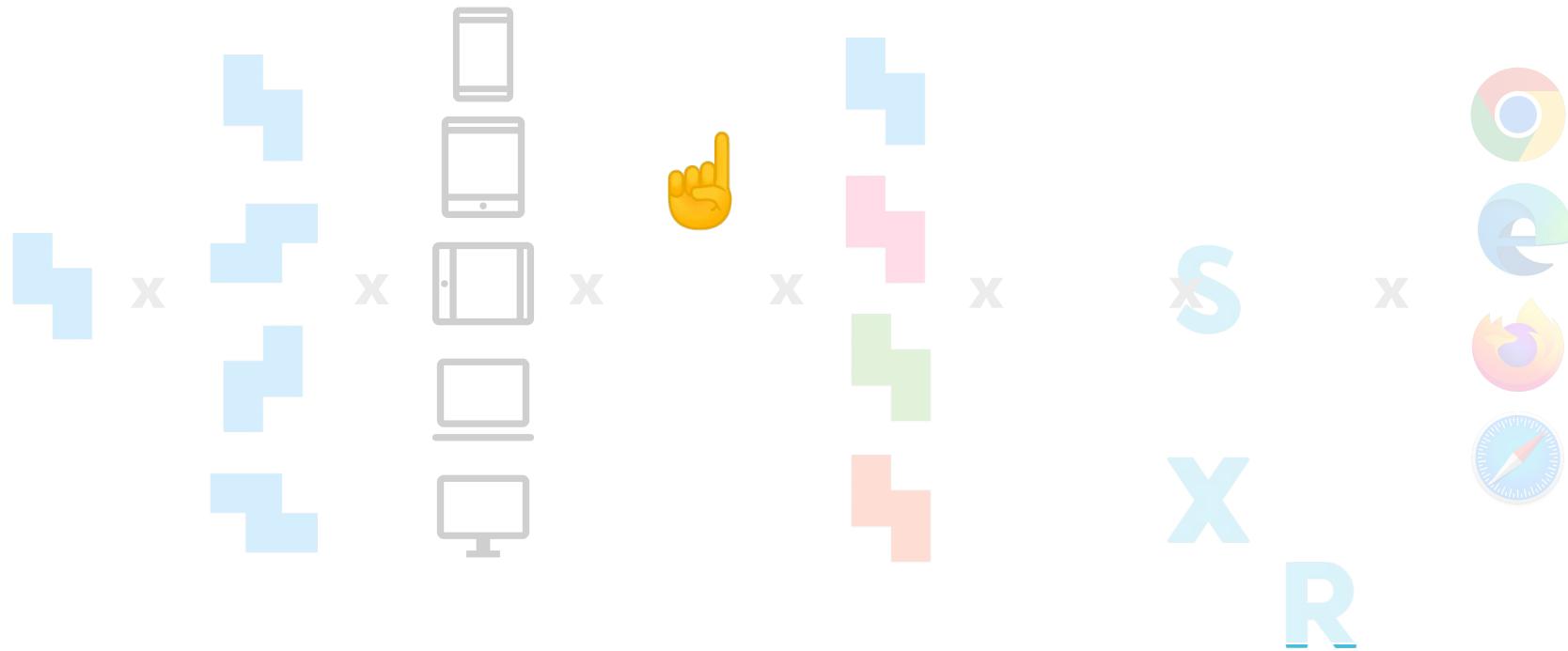
```
// .storybook/preview.js
import { LocaleProvider } from '**/LocaleProvider'
import { ThemeProvider } from '**/ThemeProvider'

export const parameters = {}

export const globalTypes = {}

export const decorators = [
  (Story, context) => {
    const { locale, theme } = context.globals
    return (
      <LocaleProvider locale={locale}>
        <ThemeProvider theme={theme}>
          <Story />
        </ThemeProvider>
      </LocaleProvider>
    )
  },
]
```





```
// .storybook/main.js
module.exports = {
  "stories": [
    " ../stories/**/*.{stories.mdx}",
    " ../stories/**/*.{stories.@(js|jsx|ts|tsx)}"
  ],
  "addons": [
    "@storybook/addon-links",
    "@storybook/addon-essentials",
    "@storybook/addon-interactions"
  ],
  "framework": "@storybook/react"
}
```

Storybook

Why Showcase Docs Integrations Community

Search docs

## Integrations

Add your integration

Integrate your tools with Storybook to connect workflows and unlock advanced features.

Search integrations

Theming components Material-ui Accessibility React Tickets Nextjs + 14 more

CATEGORIES

- Popular
- Essentials
- Code
- Data & state
- Test
- Style
- Design
- Appearance
- Organize

How to install addons Create an addon

### Popular addons



**Links** 2.3M Downloads

Link stories together to build demos and prototypes with your UI components



**Interactions** 859.9k Downloads

Automate, test and debug user interactions



**Accessibility** 930.7k Downloads

Test component compliance with web accessibility standards



**Designs** 414.1k Downloads

Storybook addon for embedding your design preview in addon panel



**Storysource** 330.4k Downloads

View a story's source code to see how it works and paste into your app



**Mock Service Worker** 192.5k Downloads

Mock API requests in Storybook with Mock Service Worker.

Add your integration

## Storybook Integrations



```
// .storybook/main.js
module.exports = {
  "stories": [],
  "addons": [
    ...,
    "@storybook/addon-a11y"
  ],
}
```



S Storybook

Find components /

EXAMPLE

- Introduction
- Button
  - Primary
  - Secondary
  - Large
  - Small
- Header
- Page

Canvas Docs

Button

A screenshot of a Storybook interface. The left sidebar shows a navigation tree under the 'EXAMPLE' section. The 'Button' category is expanded, with its first item, 'Primary', highlighted by a blue bar. The main canvas area displays a single blue button component. The top navigation bar includes tabs for 'Canvas' and 'Docs', along with various tool icons.

The screenshot shows the Storybook interface with the following details:

- Header:** Storybook
- Canvas:** Shows a blue button labeled "Button" with a red dashed border.
- Sidebar (Example section):**
  - Introduction
  - Button** (selected)
  - Primary (highlighted in blue)
  - Secondary
  - Large
  - Small
- Header (under Button):**
  - Controls (4)
  - Actions
  - Interactions
  - Accessibility (1)** (highlighted in blue)
- Accessibility Summary:** 1 Violations, 3 Passes, 0 Incomplete
- Violations:** Elements must have sufficient color contrast (checked)
- Description:** Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds
- Link:** More info...
- Code:** 1. .storybook-button
- Details:** Element has insufficient color contrast of 2.62 (foreground color: #ffffff, background color: #1ea7fd, font size: 10.5pt (14px), font weight: bold). Expected contrast ratio of 4.5:1
- Tags:** cat.color, wcag2aa, wcag143, ACT



# Storybook as a workshop environment

- [Stories](#)
- [CSF v3](#)
- [Controls](#)
- [Args](#)
- [Measure & outline](#)
- [Toolbars & globals](#)
- [Decorators](#)
- [Configure Storybook](#)
- [Integrations \(addons\)](#)

*Test*

# **Storybook as a test environment**



# Storybook Test Runner



```
// .storybook/test-runner.js

const { injectAxe, checkA11y } = require('axe-
playwright');

module.exports = {
  async preRender(page, context) {
    await injectAxe(page);
  },
  async postRender(page, context) {
    await checkA11y(page, '#root', {
      detailedReport: true,
      detailedReportOptions: {
        html: true,
      },
    })
  },
};
```



```
// package.json
{
  "scripts": {
    "test-storybook:ci": "test-storybook"
  }
}
```



ui-testing-guide-code: yarn storybook

~/Documents/chromatic/ui-testing-guide-code

```
No accessibility violations detected!
at DefaultTerminalReporter.<anonymous> (node_modules/axe-playwright/dist/reporter/defaultTerminalReporter.js:46:25)
console.log
No accessibility violations detected!
at DefaultTerminalReporter.<anonymous> (node_modules/axe-playwright/dist/reporter/defaultTerminalReporter.js:46:25)
console.log
No accessibility violations detected!
at DefaultTerminalReporter.<anonymous> (node_modules/axe-playwright/dist/reporter/defaultTerminalReporter.js:46:25)
console.log
No accessibility violations detected!
at DefaultTerminalReporter.<anonymous> (node_modules/axe-playwright/dist/reporter/defaultTerminalReporter.js:46:25)
Test Suites: 4 passed, 4 total
Tests:    15 passed, 15 total
Snapshots: 15 passed, 15 total
Time:    3.721 s, estimated 5 s
Ran all test suites.
† Done in 6.00s.
```



```
// .storybook/main.js
module.exports = {
  // ...
  addons: [
    // ...
    '@storybook/addon-coverage'
  ],
};
```



```
// package.json
{
  "scripts": {
    "test-storybook:ci": "test-storybook --coverage"
  }
}
```

```
$ node bin/test-storybook.js --coverage
PASS browser: chromium stories/atoms/Button.stories.js
PASS browser: chromium stories/pages/Page.stories.js
PASS browser: chromium stories/molecules/Header.stories.js

Test Suites: 3 passed, 3 total
Tests:       6 passed, 6 total
Snapshots:   10 passed, 10 total
Time:        3.835 s
Ran all test suites.
Coverage file (7200 bytes) written to .nyc_output/coverage.json
-----|-----|-----|-----|-----|-----|
File    | %Stmts | %Branch | %Funcs | %Lines | Uncovered Line #
-----|-----|-----|-----|-----|-----|
All files | 100 | 83.33 | 100 | 100 |
atoms     | 100 | 75 | 100 | 100 |
  Button.js | 100 | 75 | 100 | 100 | 18
molecules | 100 | 100 | 100 | 100 |
  Header.js | 100 | 100 | 100 | 100 |
pages     | 100 | 100 | 100 | 100 |
  Page.js   | 100 | 100 | 100 | 100 |

-----|-----|-----|-----|-----|-----|
:+ Done in 5.52s.
```

“

*Writing your component  
UI tests in Jest can feel a  
little like stabbing at your  
components with a  
blindfold on.*

- Michael Chan



# Play functions/interactions



```
// .storybook/main.js
module.exports = {
  stories: [],
  addons: [
    // Other Storybook addons
    '@storybook/addon-interactions', // 🎨 Register the addon
  ],
}
```



```
// Button.stories.jsx
import { expect } from '@storybook/jest'
import { userEvent, waitFor, within } from '@storybook/testing-library'
...
export const Primary = {
  args: {
    label: "Button"
  },
  play: async ({args, canvasElement}) => {
    const canvas = within(canvasElement)

    await userEvent.click(canvas.getByRole('button'))

    await waitFor(() => expect(args.onClick).toHaveBeenCalled())
  }
}
```

The screenshot shows the Storybook interface for a 'Button' component. The top navigation bar includes tabs for 'Canvas' (which is active), 'Docs', and various icons for search, refresh, and export. A search bar at the top left contains the placeholder 'Find components /'. On the left, a sidebar titled 'EXAMPLE' lists components: 'Introduction' (closed), 'Button' (expanded), 'Primary' (selected and highlighted in blue), 'Secondary', 'Large', 'Small', 'Header', and 'Page'. The main content area displays a large blue button labeled 'Button'. Below the button, there are four tabs: 'Controls (4)', 'Actions (1)', 'Interactions (2)' (which is active and highlighted in blue), and 'Readme'. Under the 'Interactions (2)' tab, two test snippets are shown, both of which have passed: 

```
✓ userEvent.click(within(<div#root>).getByRole("button"))
```

 and 

```
✓ expect(action(onClick)).toHaveBeenCalled()
```

.



```
// Button.stories.jsx
import { expect } from '@storybook/jest'
import { userEvent, waitFor, within } from '@storybook/testing-library'
...
export const Primary = {
  args: {
    label: "Button",
  },
  play: async ({args, canvasElement}) => {
    const canvas = within(canvasElement)

    await userEvent.click(canvas.getByRole('button'))

    await waitFor(() => {
      expect(args.onClick).toHaveBeenCalled()
      expect(canvas.getByRole('button')).toBeDisabled()
    })
  }
}
```

The screenshot shows the Storybook interface for a 'Button' component. The top navigation bar includes 'Canvas' (which is selected), 'Docs', and various tool icons. A search bar at the top left says 'Find components /'. On the left, a sidebar titled 'EXAMPLE' lists components like 'Introduction', 'Button' (selected), 'Primary' (selected), 'Secondary', 'Large', 'Small', 'Header', and 'Page'. The main area shows a blue button labeled 'Button'. Below it, a card displays test results for interactions:

Controls (4)	Actions (2)	Interactions (2)	Readme
✓ userEvent.click(within(<div#root>).getByRole("button"))			
✓ expect(action(onClick)).toHaveBeenCalled()			
		✗ expect(within(<div#root>).getByRole("button")).toBeDisabled()	
		expect(element).toBeDisabled()	

The failed interaction test is highlighted with a red error icon and a yellow background. The error message states: 'Received element is not disabled:' followed by the HTML code for the button.

# **Stories in tests**



```
// Form.test.js

import { render, fireEvent } from '@testing-library/react';

import { InvalidForm } from './LoginForm.stories'; // ➔ Import story

it('Checks if the form is valid', () => {
  // 🚫 Use in test
  const { getByTestId, getByText } = render(<InvalidForm {...InvalidForm.args} />);

  fireEvent.click(getByText('Submit'));

  const isFormValid = getByTestId('invalid-form');
  expect(isFormValid).toBeInTheDocument();
});
```



chromatic



**BASELINE HISTORY** ⓘ

Build 2 • 791a1d

Discussions

hichroma/acme

Builds PRs Library Manage

## Build 2

Initiated 4 hours ago

148 Stories 1m 7s Build time Browsers Add Chromatic Pull request 81e4b4 feature-branch

TESTS Visual Interactions TurboSnap

167 Tests ① 4 Changes Build 1 Ancestor 0 Accepted 0 Denied 4 Unreviewed

Review changes

Changes	Status	Type	Actions
Default	Unaccepted	Visual	✓ ✖
Low data	Unaccepted	Visual	✓ ✖
Last hour	Unaccepted	Visual	✓ ✖
Last week	Unaccepted	Visual	✓ ✖

Find by name View Storybook

hichroma/acme

Build 2  
Initiated 4 hours ago

23 Components | 148 Stories

148 Stories | 1m 7s Build time | Add Chromatic Pull request | 81e4b4 feature-branch

TESTS | Visual | Interactions | TurboSnap

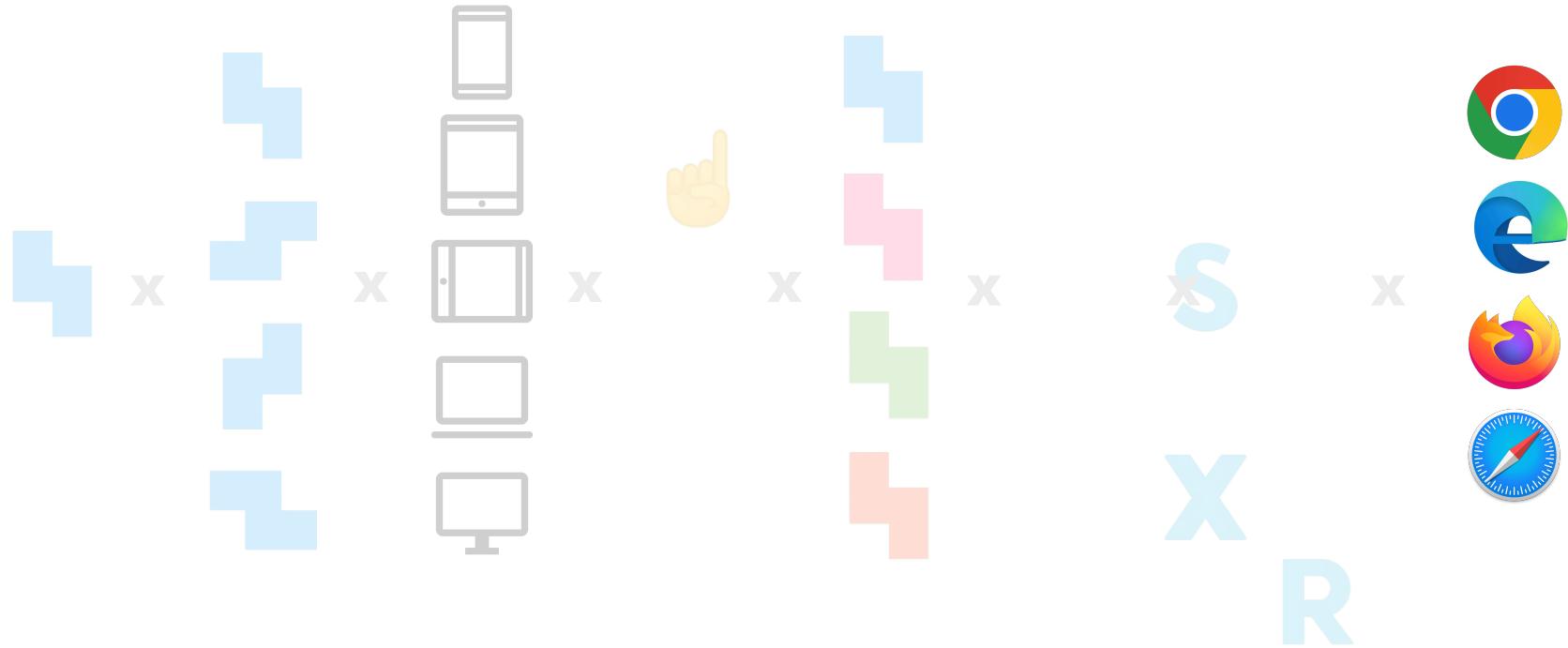
167 Tests | 4 Changes | Build 1 Ancestor | 4 Accepted | 0 Denied | 0 Unreviewed | ✓: Batch | Review changes

Changes	Status	Type	Actions
Histogram	--	--	✓
Default	Accepted	Visual	↶
Low data	Accepted	Visual	↶
DatePicker	--	--	✓
Last hour	Accepted	Visual	↶
Last week	Accepted	Visual	↶



**UI Tests** – 4 changes accepted as baselines





## TESTS



**Visual tests:** Enabled

[Disable](#)

Automatically check for UI bugs in each component every commit to pinpoint UI bugs. Snapshots are captured in Chrome by default. Select additional browsers to generate new baselines for each story on the next build. [Learn more »](#)



✓ Chrome



✓ Firefox



ⓘ IE 11

# **Storybook as a test environment**

- Test runner
- A11y tests
- Test coverage
- Play functions/interactions
- Chromatic

*Review & Document*

# **Storybook as a communication tool**

• PR check

Review now ▾

739f420

### REVIEWERS

8c2a5f3



Assign reviewers

aaa3477



Assign yourself

cf897de



### PARTICIPANTS



Gert Hengeveld



## B A S E L I N E H I S T O R Y ?

### Build 2 • 791a1d

Most recent build. There is an unaccepted change, so Build 1 is still the baseline.

### Build 1 • 791a1d current baseline Auto-accepted

## Discussions



Comment on this change

Comment

Comment + deny

## D O M D I F F ?

```
... @@ -1,5 +1,5 @@
```

```
1 import React, { PropTypes } from 'react';
2 - import { Match } from 'react-router';
3
```

```
1 import React, { PropTypes } from 'react';
2 + import { Route } from 'react-router-dom';
3
```

hichroma/acme

Onboarding improvements v2

Unmerged Merging feature-branch into master • View GitHub PR

Builds PRs Library Manage

Activity Components Changeset 37 PR check Review now

Build 6 3 jobs run 2 days ago x8219ce  
Build 7 3 jobs run 12 hours ago jk1h2kd

Histogram: Loading View component Resolve

Tom Coleman 10 days ago I think maybe the browser panels should be expandable/retractable and start retracted if they are unchanged, maybe just leaving the header?

Zoltan Olah 1 day ago Looks good for the most part but the hover transition feels slow. Can we speed it up?

Add a reply

Andrew assigned James to review 12 hours ago 1ydicie

Build 18 3 jobs run 3 hours ago

James approved this PR 2 hours ago

Micah assigned Tom to review 20 minutes ago

Onboarding improvements v2 Complete UI checklist below to pass the GitHub PR check. Learn more » View PR on GitHub

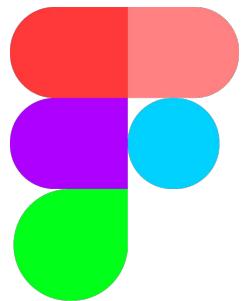
1 pending reviewer All reviewers must accept this PR to pass the status check Tom Coleman (you) Pending Unassign

1 unresolved discussions Resolve all

37 Changes 1 Discussion 174 Stories

REVIEWERS James Adebayo ✓ Tom Coleman

PARTICIPANTS Micah Parillo Sarah Khan Andrew Lee



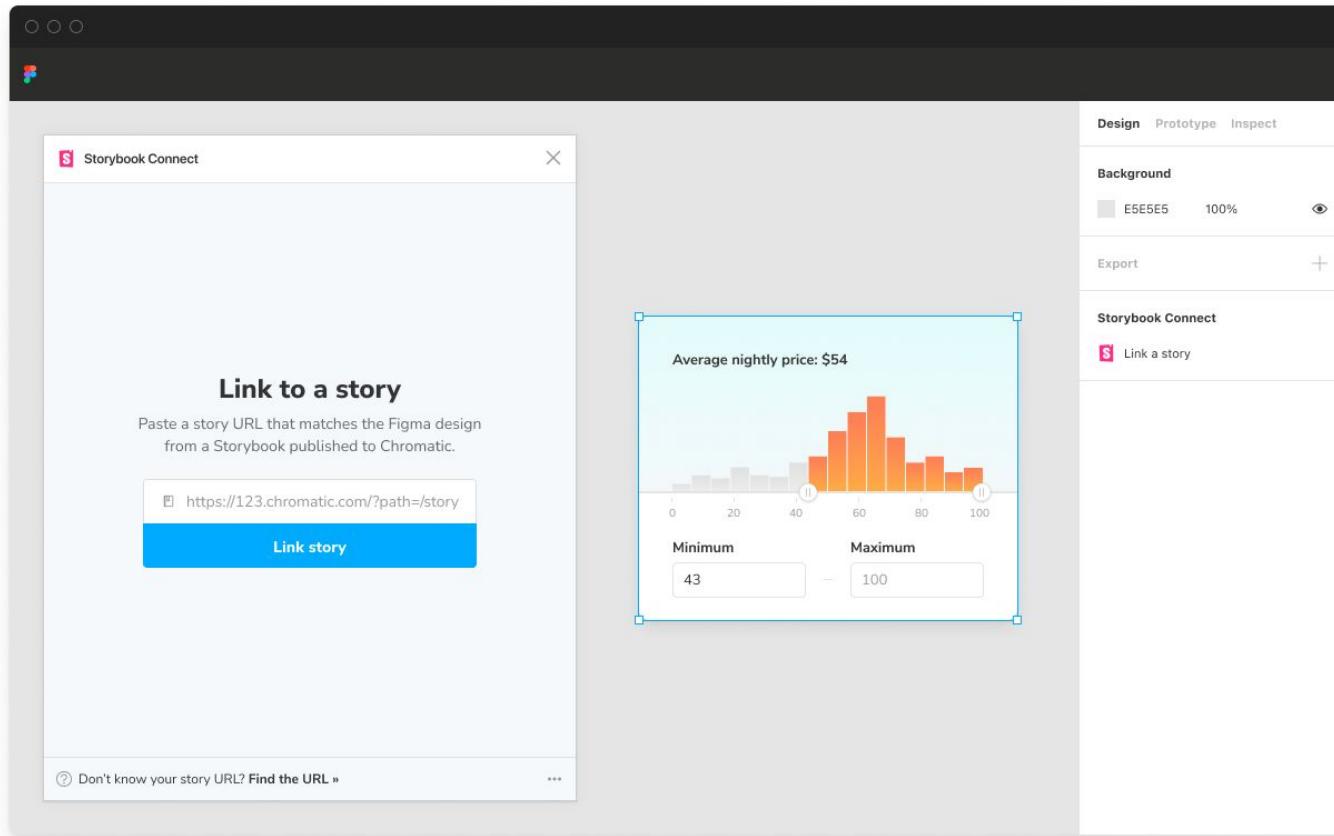
The screenshot displays two side-by-side interfaces for a UI component named "Button Primary".

**Top Interface (Storybook):**

- A top navigation bar includes "STORYBOOK" and a "Filter" dropdown.
- A sidebar on the left lists categories: "Button" (expanded), "With Figma" (selected), "With Figma No Fullscreen Option", "Without Figma", and "App" (expanded) with "My App" listed under it.
- The main content area shows a single "Button Primary" component.

**Bottom Interface (Figma):**

- A top navigation bar includes "ACTION LOGGER" and "FIGMA".
- The main content area is titled "Figma".
- Three instances of the "Button Primary" component are displayed as white rectangular cards with blue rounded corners and the text "Button Primary" in the center.
- At the bottom of the Figma interface, there is a footer note: "Button Primary, edited a month ago".



Storybook Connect for Figma

○ ○ ○



Grayscale/Gray-600 mediumDark

Grayscale/Gray-500 (unused?)

Grayscale/Gray-900 darkest

See all 5 styles

Linear 100%

FFFFFF 100%

000000 10%

See all 5 colors

Effects



Export



Storybook Connect

View story

Open story in browser



Design Prototype Inspect

Background

E5E5E5 100%



Export

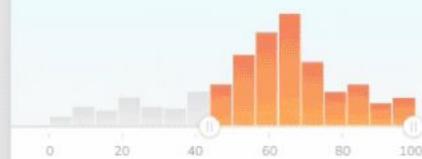


Storybook Connect

View story

Open story in browser

Average nightly price: \$54



Minimum

43

Maximum

100

○ ○ ○

Storybook Connect

🔍 🔍 🔍 | ⚒ ⚒ ⚒ | ⚒ ⚒ ⚒

Export +

Background

E5E5E5 100% ⚒

Plugin

View story

Open story in browser

Groupitem: Default

...

The screenshot shows the Storybook Connect extension integrated into a design application. On the left, the Storybook Connect sidebar displays a chart component with a blue area plot and a light blue gradient fill. Below the chart are the author's profile picture ('domyen'), a view count of '15k', and a comment count of '7'. The right side of the interface is a design tool's panel, featuring a preview of the same chart component. The preview is surrounded by a purple selection frame. The top right of the panel has tabs for 'Design', 'Prototype', and 'Inspect'. The 'Background' section shows a hex color code E5E5E5 and a 100% opacity setting. The 'Plugin' section includes links to 'View story' and 'Open story in browser'. At the bottom, there are status indicators for 'Groupitem: Default' and an ellipsis menu.

# **Storybook for Documentation**

The screenshot shows the Storybook interface for a UI component library. The left sidebar lists components under 'EXAMPLE' (Introduction, Button, Header, Page), with 'Button' selected. The main area displays the 'Button' story, which is described as a 'Primary UI component for user interaction'. Below the description is a preview card showing a blue button labeled 'Button' and a 'Show code' button. A detailed configuration table follows, listing the properties of the 'Primary' button component.

Name	Description	Default	Control
<b>primary</b>	is this the principal call to action on the page? boolean	false	<input checked="" type="radio"/> False <input type="radio"/> True
<b>label*</b>	Button contents string	-	<input type="text" value="Button"/>
<b>backgroundColor</b>	What background color to use string	-	<input type="button" value="Choose color..."/>
<b>size</b>	How large should the button be? "small" "medium" "large"	"medium"	<input type="radio"/> small <input type="radio"/> medium <input type="radio"/> large
<b>onClick</b>	Optional click handler (( ) => void)	-	-

**Stories**  
**Secondary**

Below the stories section is a preview card showing a blue button labeled 'Button'.

```
☰ Button.stories.mdx U X  
stories > ⌂ Button.stories.mdx  
1 import { Canvas, Meta, Story } from '@storybook/addon-docs'  
2 import { Button } from './Button'  
3 import * as stories from './Button.stories.tsx'  
4  
5 <Meta title="Example/Button" component={Button} />  
6  
7 # Button  
8  
9 <Canvas isExpanded>  
10 | <Button label="Button" primary/>  
11 </Canvas>  
12  
13 ## Usage  
14  
15 Lorem ipsum dolor sit amet, consectetur adipiscing elit...  
16  
17 <Story name="Primary" story={stories.Primary}/>
```

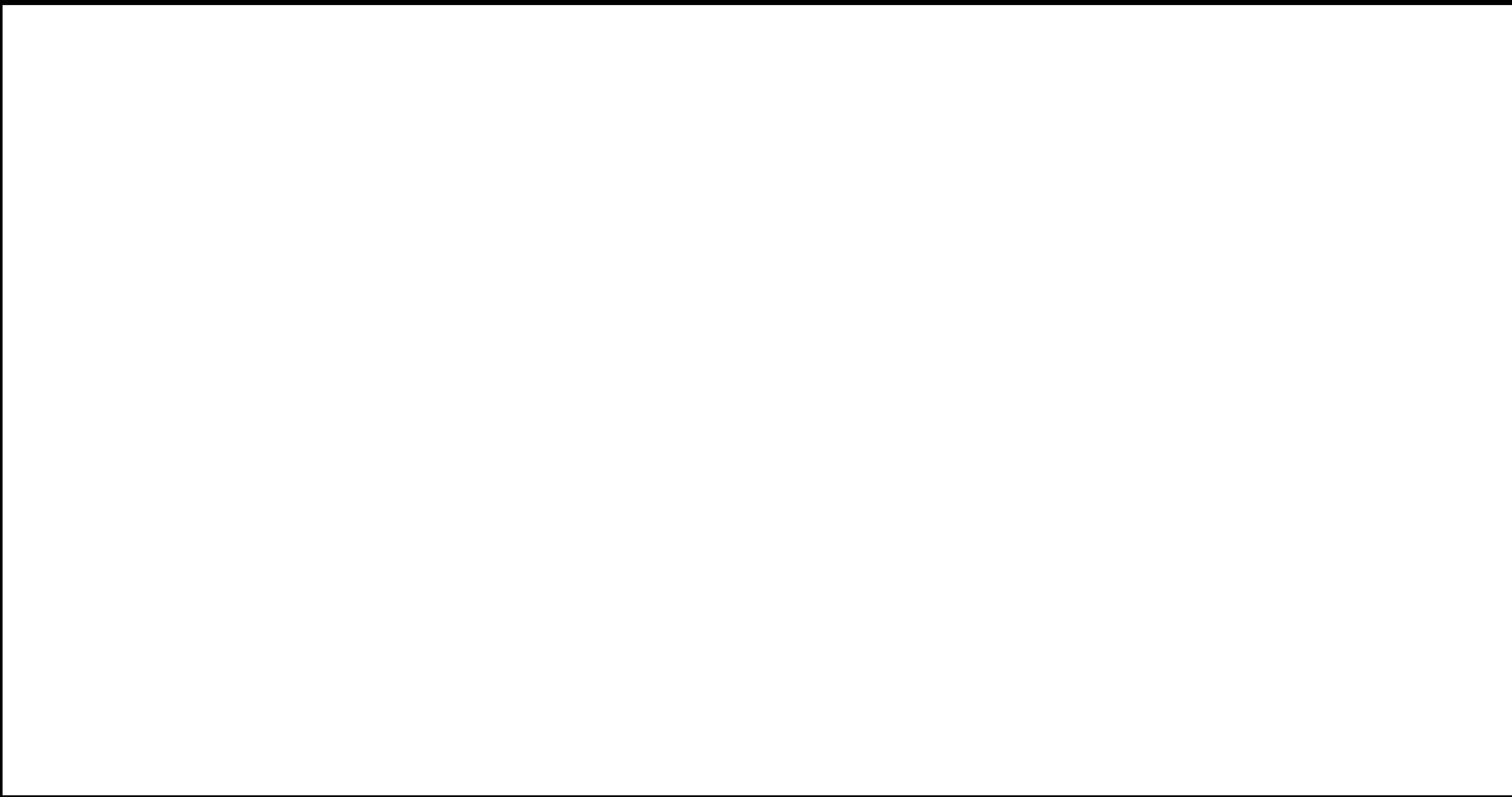
The screenshot shows the Storybook interface for a component library. The top navigation bar includes the Storybook logo, a three-dot menu, and tabs for 'Canvas' and 'Docs'. A search bar labeled 'Find components' is present. On the left, a sidebar titled 'EXAMPLE' lists components: 'Button' (selected), 'Primary' (highlighted in blue), 'Secondary', 'Large', 'Small', 'Introduction', 'Header', and 'Page'. The main content area displays the 'Button' component's documentation. It features a large bold heading 'Button' and a preview section with a blue button labeled 'Button'. Below the preview is a code snippet in a dark box:

```
<Button label="Button" primary />
```

With a 'Hide code' button and a 'Copy' button. The 'Usage' section contains placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ut orci nec orci elementum posuere. Lorem ipsum dolor sit amet, consectetur adipiscing elit.' At the bottom is another blue button labeled 'Button'.

**What features to use?  
What stories to write?**

# **What components to write?**



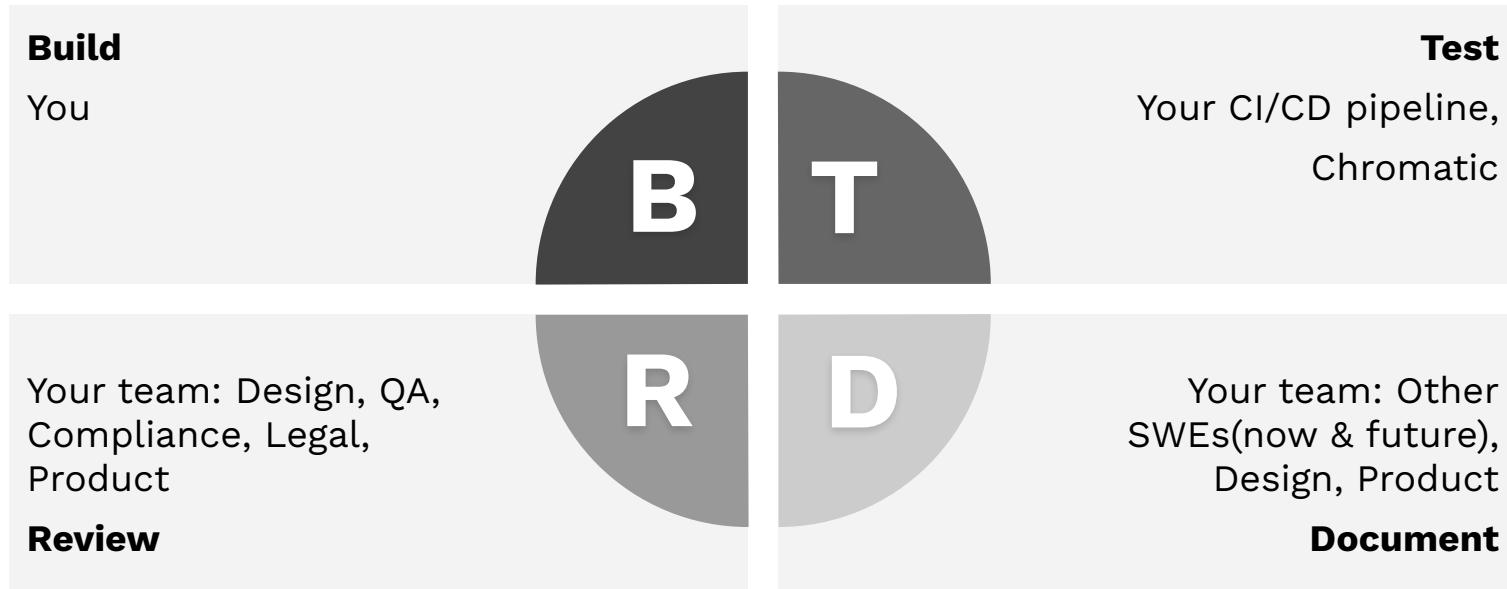
# **Storybook as a communication tool**

- [Chromatic](#)
- [Figma plugin](#)
- [DocsPage](#)
- [MDX](#)

*Case study*

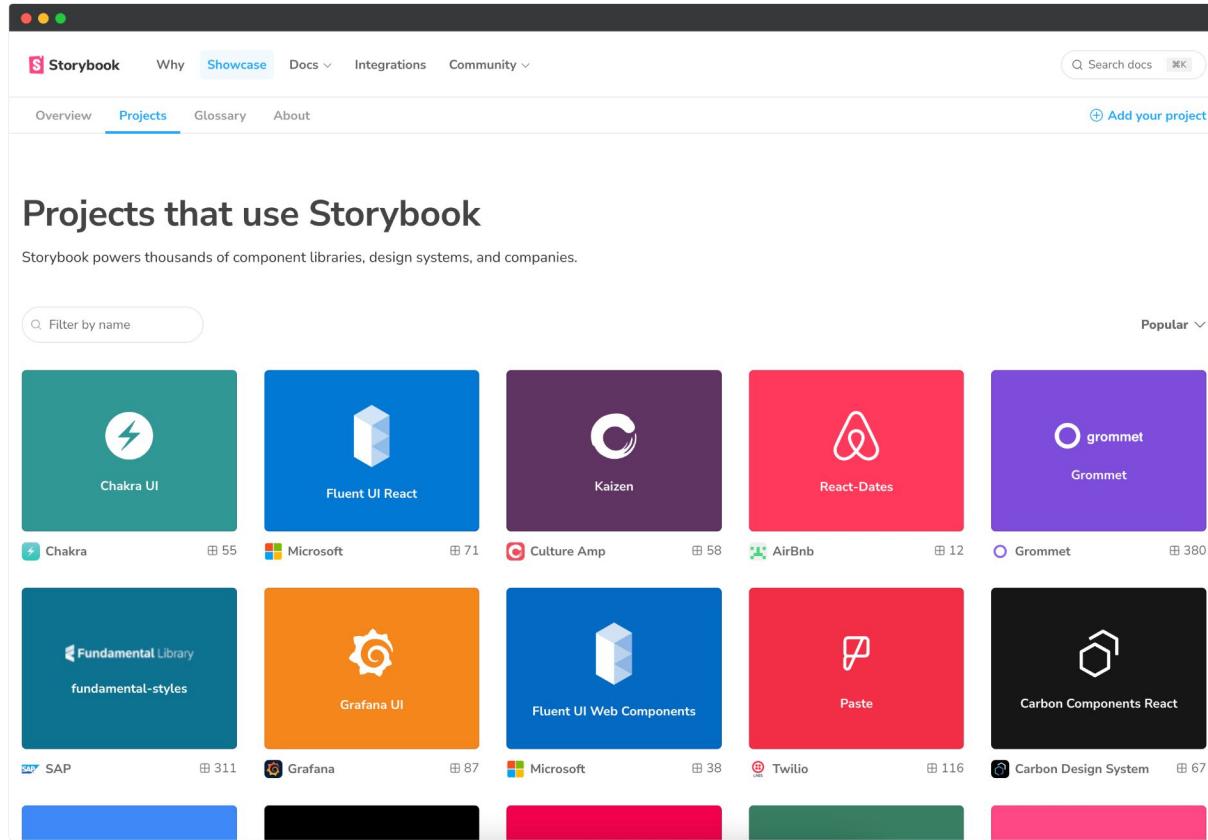
# Storybook for applications

# Storybook users across the SDLC



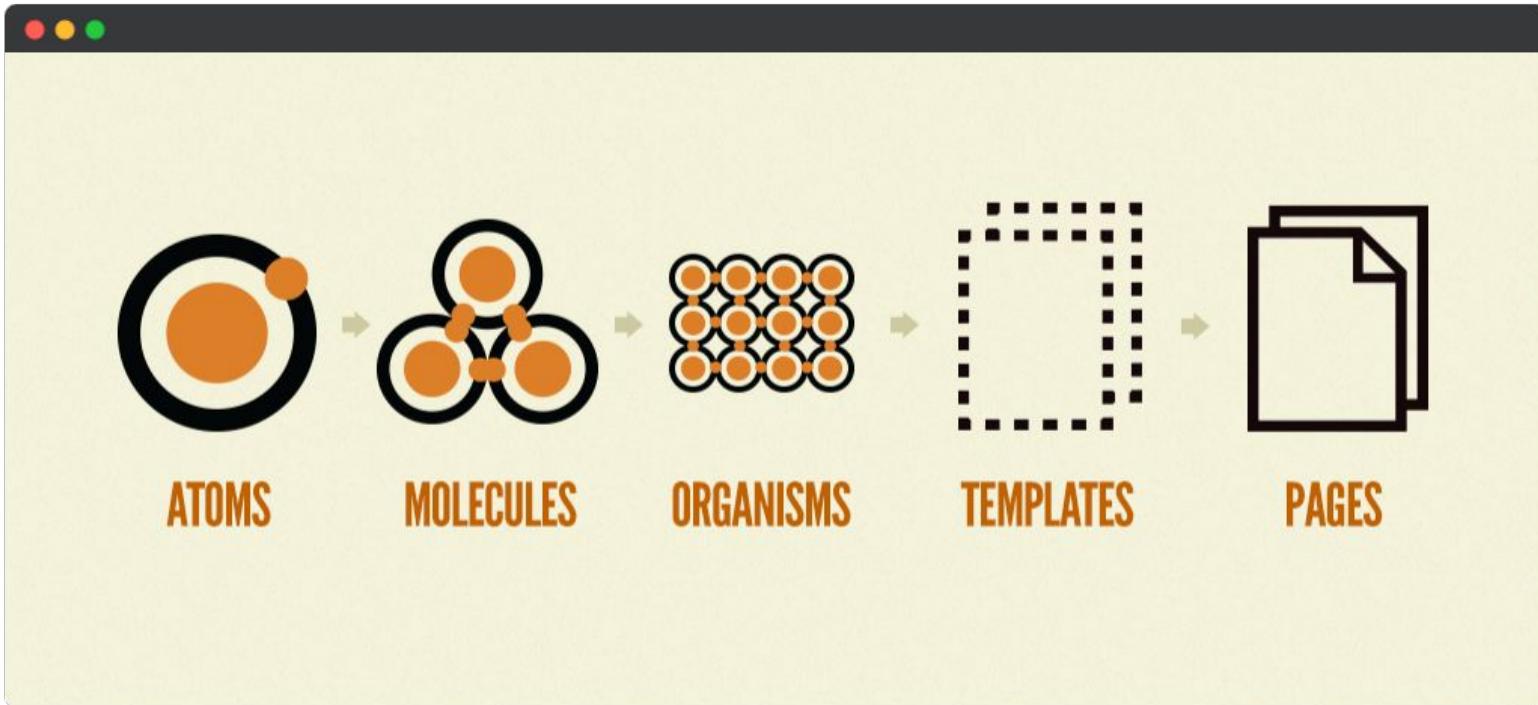
# **Communicating to yourself**

Stories

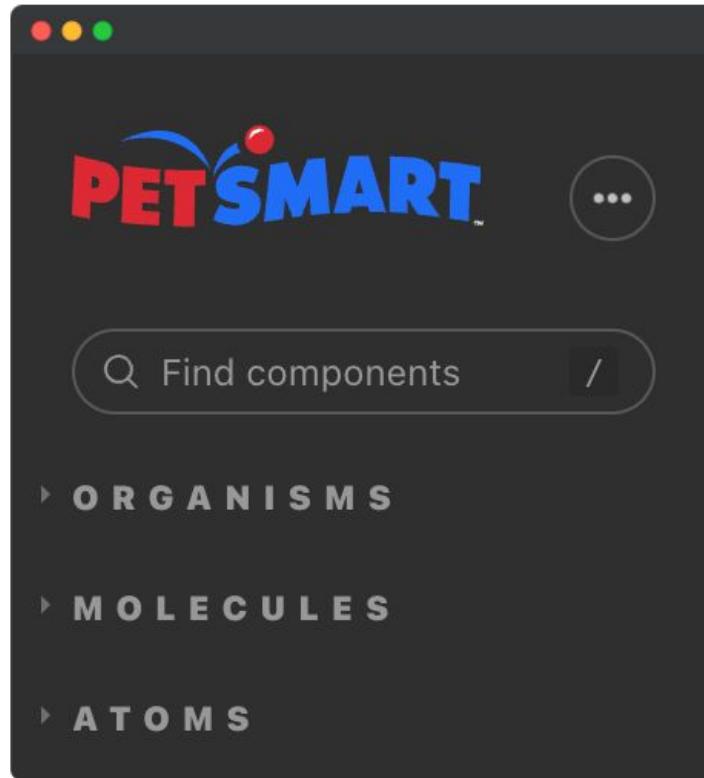
A screenshot of the Storybook website's "Projects" page. The page features a header with navigation links like "Storybook", "Why", "Showcase", "Docs", "Integrations", and "Community". A search bar is located in the top right. Below the header, a sub-navigation bar includes "Overview", "Projects" (which is underlined in blue), "Glossary", and "About". A button "+ Add your project" is also present. The main content area is titled "Projects that use Storybook" and contains a sub-header "Storybook powers thousands of component libraries, design systems, and companies.". A search input field "Filter by name" is available. To the right, a "Popular" dropdown menu is shown. The main content consists of a grid of ten project cards, each with a logo, name, and a small icon. The projects listed are Chakra UI, Fluent UI React, Kaizen, React-Dates, grommet, fundamental-library, Grafana UI, Fluent UI Web Components, Paste, and Carbon Components React.

Project	Icon	Description	Count
Chakra UI		Chakra	55
Fluent UI React		Microsoft	71
Kaizen		Culture Amp	58
React-Dates		AirBnb	12
grommet		Grommet	380
fundamental-library		fundamental-styles	311
Grafana UI		Grafana	87
Fluent UI Web Components		Microsoft	38
Paste		Twilio	116
Carbon Components React		Carbon Design System	67

[Storybook Component Encyclopedia](#)



Atomic Design | Brad Frost



PETSMART Storybook

# **Communicating to your CI/CD pipeline**

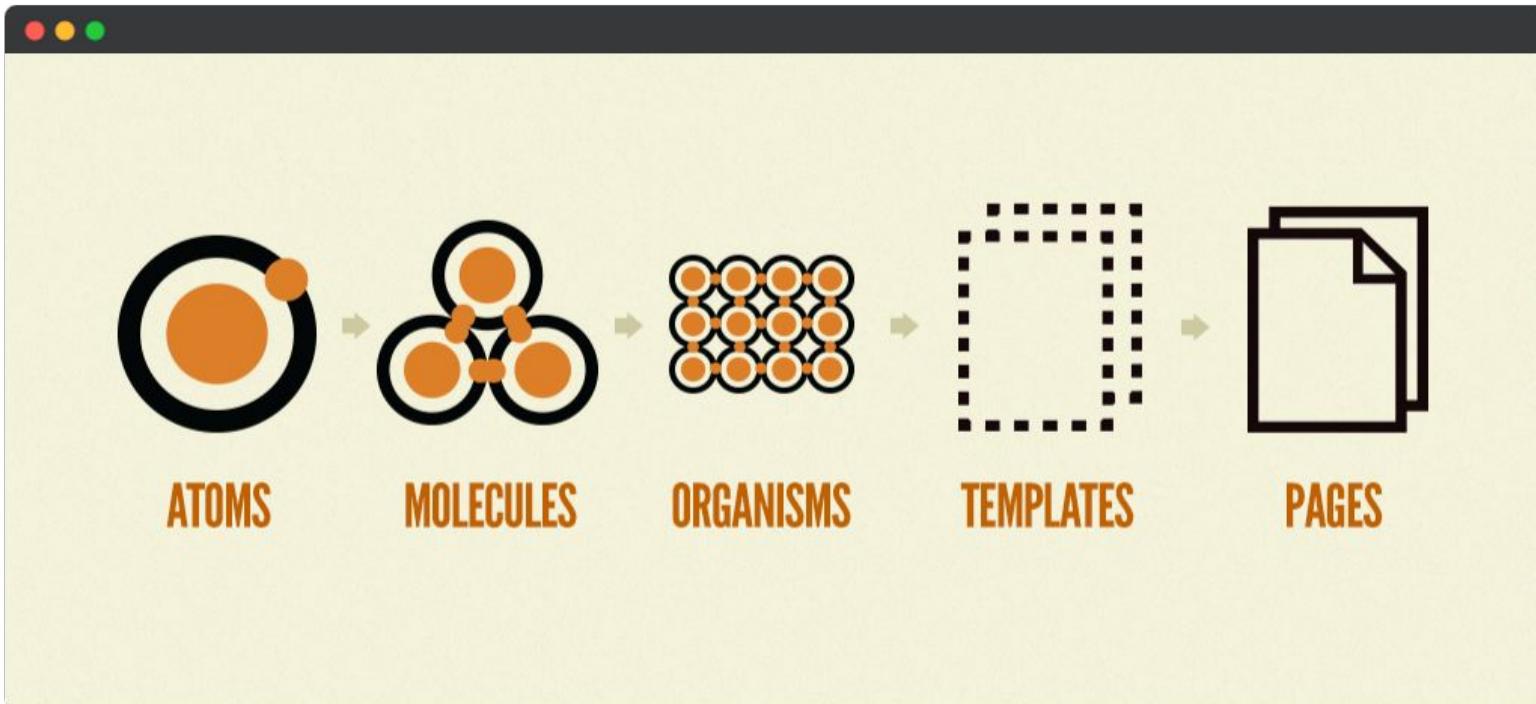
Chromatic

The screenshot shows the Shopify Storybook interface. On the left, a sidebar lists various components under categories like 'Playground' and 'All Components'. The 'Details Page' component is currently selected. The main content area displays a component's configuration page for a 'Success badge'. The page includes fields for 'Title' (set to 'The North Face Ventrix Active Trail Hybrid Hoodie - Men's'), 'Description' (containing a detailed product description), and 'Media' (with a placeholder for adding files). To the right, there are sections for 'Organization' (Product type: Keyboard, Vendor: Keyboard), 'Collections', and 'Tags'.

Shopify Storybook

# **Communicating to your team**

Other Frontend Engineers



# **Communicating to your team**

Backend Engineers

```
// YourPage.jsx
import { PageLayout } from './PageLayout';
import { DocumentHeader } from './DocumentHeader';
import { DocumentList } from './DocumentList';

function useFetchData() {
    // fetch data from external API
}

export function DocumentScreen() {
    const { status, data } = useFetchData();

    if (status === 'loading') {
        return <p>Loading ... </p>;
    }
    if (status === 'error') {
        return <p>There was an error fetching the data!</p>;
    }
    return (
        <PageLayout user={data.user}>
            <DocumentHeader document={data.document} />
            <DocumentList documents={data.subdocuments} />
        </PageLayout>
    );
}
```



```
// YourPage.stories.jsx
import React from 'react';
import { rest } from 'msw';
import { DocumentScreen } from './YourPage';

export default {
  title: 'DocumentScreen',
  component: DocumentScreen,
};

const TestData = {
  // The mocked data that will be used in the story
};

const PageTemplate = () => <DocumentScreen />

export const MockedSuccess = PageTemplate.bind({});
MockedSuccess.parameters = {
  msw: [
    rest.get('https://your-restful-endpoint/', (_req, res, ctx) => {
      return res(ctx.json(TestData));
    }),
  ],
};
```

```
// YourPage.jsx
import { PageLayout } from './PageLayout';
import { DocumentHeader } from './DocumentHeader';
import { DocumentList } from './DocumentList';

function useFetchData() {
    // fetch data from external API
}

export function DocumentScreen() {
    const { status, data } = useFetchData();

    if (status === 'loading') {
        return <p>Loading ... </p>;
    }
    if (status === 'error') {
        return <p>There was an error fetching the data!</p>;
    }
    return (
        <PageLayout user={data.user}>
            <DocumentHeader document={data.document} />
            <DocumentList documents={data.subdocuments} />
        </PageLayout>
    );
}
```



```
// YourPage.jsx
import { DataDashboard } from './DataDashboard';

function useFetchData() {
    // fetch data from external API
}

export function DocumentScreen() {
    const { status, data } = useFetchData();

    if (status === 'loading') {
        return <p>Loading ... </p>;
    }
    if (status === 'error') {
        return <p>There was an error fetching the data!</p>;
    }
    return (
        <DataDashboard data={data}>
    );
}
```

# **Communicating to your team**

Design & QA

**FORMS**

- Checkbox
- FormErrorState
  - Default
  - Pure Multiple Errors
- Pure Multiple Errors
  - Pure Multiple Errors
  - Suppressed
- Input
- Radio
- Select
- Textarea

**CLIPBOARD**

- Clipboard
- ClipboardCode
- ClipboardIcon
- ClipboardInput

**MARKETING**

- Breadcrumb

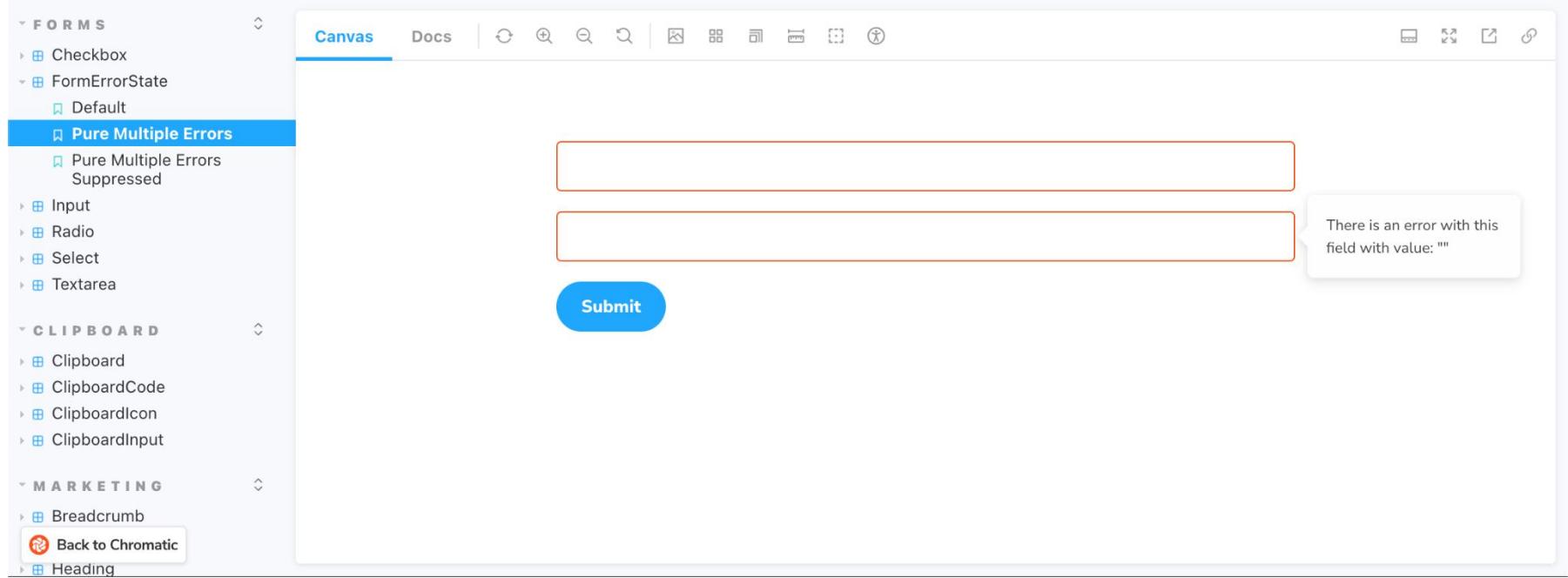
[!\[\]\(bc3f7b6ffdefa923de5918f74d1c2952\_img.jpg\) Back to Chromatic](#)

[!\[\]\(dd76831d71f8ae23a111248466c7002d\_img.jpg\) Heading](#)

Canvas Docs |          

Canvas

There is an error with this field with value: ""



Storybook Design System

# **Communicating to your team**

Product

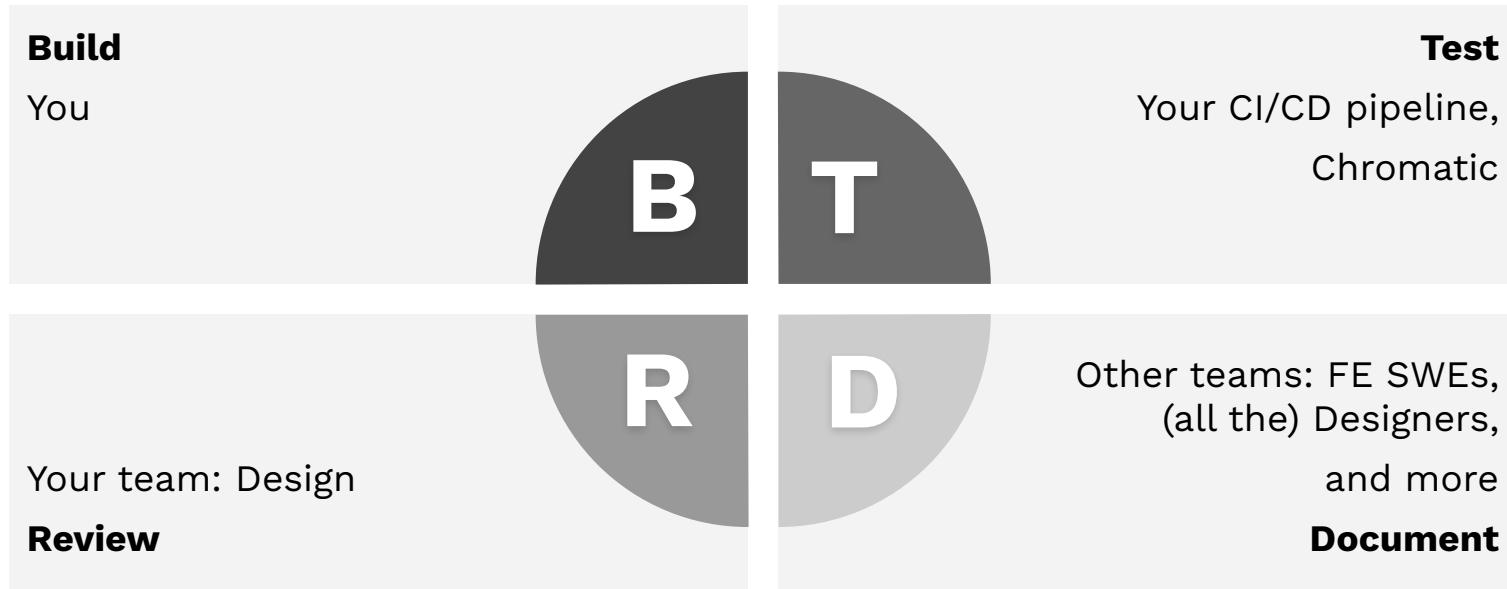
# **Storybook for applications**

- Mock Service Worker addon

*Case study*

# Storybook for design systems

# Storybook users across the SDLC



# **Communicating to your CI/CD pipeline**

Chromatic

Workday / canvas-kit Public

Watch 24 Fork 163 Star 256

Code Issues 107 Pull requests 7 Discussions Actions Projects 1 Security ...

master canvas-kit / modules / react / button / stories / Go to file Add file ...

3 authors chore: Re organize storybook (#1943) ... 7 hours ago History

..

button	chore: Re organize storybook (#1943)	7 hours ago
visual-testing	chore: Re organize storybook (#1943)	7 hours ago
stories_ToolbarButtons.tsx	chore: Re organize storybook (#1943)	7 hours ago
stories_deprecatedButton.tsx	chore: Re organize storybook (#1943)	7 hours ago
tsconfig.json	refactor(popup): Convert Popup to a compound component (#1065)	last year

## Canvas-kit Storybook

# **Communicating to the masses**

Developers

EXAMPLES

- CookieBanner
- Forms
  - Controlled Inputs Using Formik
  - Text Input With Formik**
  - Select Using Formik
  - Select With Formik
  - Uncontrolled Inputs Using React Hook Form
  - Text Input With React Hook Form
- GlobalHeader
- PageHeader
- SidePanel With Overlay

Canvas Docs | ⌂ ⌂ ⌂ ⌂ | ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

Email \*

 ⋮

Password \*

 ⋮ 

Password should be of minimum 8 characters length

Submit

COMPONENTS

- Buttons
  - Custom Styles
  - Delete
  - Primary
  - Primary Inverse
  - Secondary
  - Secondary Inverse
  - Tertiary
  - Tertiary Inverse
- Action Bar
- Deprecated
- Hyperlink
- Segmented Control
- Toolbar
- Containers
- Indicators
- Inputs
- Navigation
- Popups
- Text
  - Body Text
    - Basic
    - Heading

Canvas Docs

Large Body Text

Medium Body Text

Small Body Text

Controls Readme Story

```
1 import React from 'react';
2 import {BodyText} from '@workday/canvas-kit-react/text';
3
4 export const Basic = () => (
5   <>
6     <BodyText size="large">Large Body Text</BodyText>
7     <BodyText size="medium">Medium Body Text</BodyText>
8     <BodyText size="small">Small Body Text</BodyText>
9   </>
10 );
```



## MONOSPACED



🔍 Press "/" to search...

☰ Introduction

☰ Color

☰ **Iconography**

☰ Typography

☰ Components

☰ Blockquote

☰ Button

☰ Code

☰ Footer

☰ Grid

☰ Group

☰ Heading

☰ Icons

☰ List

☰ Logomark

☰ Logotype

☰ Markdown

☰ Masthead

☰ Paragraph

☰ Poster

# Iconography



Monospaced Storybook

Docs

# Changelog

All notable changes to this project will be documented in this file. See [Conventional Commits](#) for commit guidelines.

## v8.3.6 (2023-01-09)

### Infrastructure

- fix: Storybook sidebar highlight styling ([#1975](#)) ([@jaclynjessup](#))

## v8.3.5 (2023-01-09)

### Documentation

- chore: Re organize storybook ([#1943](#)) ([@mannycarrera4](#), [manuel.carrera](#), [@jaclynjessup](#)) We've updated our categories and organization on storybook to better surface information.

## v8.3.4 (2023-01-09)

### Testing

- test(tooltip): Add delay to fix flaky placement tests on tooltips ([#1968](#)) ([@mannycarrera4](#), [manuel.carrera](#))

The screenshot shows the CultureAmp Storybook interface. At the top, there's a navigation bar with three colored dots (red, yellow, green) on the left and icons for search, refresh, and other navigation on the right. Below the navigation bar is a header with tabs: 'Canvas' (disabled), 'Docs' (selected and highlighted in blue), and several other icons. On the far right of the header are zoom and orientation controls.

The main content area has a sidebar on the left containing a tree view of components under two sections: 'DEPRECATED' and 'TAILWIND'. Under 'DEPRECATED', the 'Filter Menu' component is expanded, showing its sub-components: 'Simple Filter (Kaizen Site Demo)', 'Default (Empty)', 'Default with children (Simple filter)', 'Default with children (Advanced filter)', 'Auto hide behaviours', 'Hero Card', and 'Text'. Under 'TAILWIND', there is a single entry: 'Filter Menu'.

The main content area displays the 'Filter Menu' component. It features a large bold heading 'Filter Menu'. Below it is a warning message: '⚠ This component is deprecated. No further changes will be made to it as it will be superseded by <FilterMultiSelect> from @kaizen/select.' A code snippet below the warning shows the import statement: 'Import { FilterMenuButton } from "@kaizen/draft-filter-menu-button"'.

At the bottom of the content area is a search bar with three input fields and a small icon on the right.

CultureAmp Storybook

**REAVIZ**

REAVIZ is a modular chart component library that leverages React natively for rendering the components while using D3js under the hood for calculations. The library provides an easy way to get started creating charts without sacrificing customization ability.

If you are looking for workflow/diagram graphs, checkout [reaflow](#).

If you are looking for webgl-based network graphs, checkout [reagraph](#).

build passing npm v13.3.2 downloads 16K/week license apache2  
minzipped size 213.3 KB Stars 743 discord 8 online

Reaviz Storybook

The screenshot shows a Storybook interface with a dark theme. At the top, there are three circular icons (red, yellow, green) followed by the text "Polaris viz". To the right of the title are three small circular buttons with ellipses, a search bar containing "Find components /", and a "Docs" tab which is currently selected. Below the header, the page title "Contributor Covenant Code of Conduct" is displayed in large, bold, white font. On the left side, there is a sidebar with a tree view of the project structure:

- Intro
- SHARED
  - POLARIS-VIZ
  - POLARIS-VIZ-NATIVE
- CONTRIBUTING
  - README
  - Code of Conduct
  - Creating Releases
  - Local Development
  - Polaris Viz Core
  - Character Widths File
  - Sandbox

The "Code of Conduct" item is highlighted with a blue background. The main content area contains two sections: "Our Pledge" and "Our Standards".

## Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

## Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people

[Polaris viz Storybook](#)

# **Communicating to the masses**

Design, et. al

The screenshot shows a Mac OS X-style window titled "Vibe Design system". The left sidebar contains a navigation menu with sections like "FOUNDATIONS" (Accessibility, Colors, Motion, Round Corners, Shadow, Spacing, Typography) and "BUTTONS" (Button, Button Group). The "Colors" section is currently selected and highlighted with a purple bar. The main content area has a header "Docs" and "Sandbox" with icons for "Find components" and "Search". Below the header is a large image of a wireframe grid with abstract shapes and a purple "p" tag. The main title "Colors" is displayed in a large, bold, dark font. A bulleted list of content items follows:

- Overview
- Color keys
- Usage and examples
- Up next
- Feedback

Vibe Design System Storybook

The screenshot shows a dark-themed web application interface. At the top, there's a header with three colored window control buttons (red, yellow, green) on the left, followed by the REAVIZ logo (an eye icon) and a three-dot menu icon. To the right of the logo is a search bar containing the placeholder "Find components" and a forward slash. Below the header, the word "Docs" is underlined in blue, indicating the current section. On the far right of the header is a full-screen icon.

The main content area has a sidebar on the left containing a navigation tree. The tree includes sections for "DOCS" (with "Intro", "Getting Started" expanded to show "Data Shapes", "Developing", and "Quick Start"), "Why" (which is selected and highlighted in blue), "Chart Types", "Utils", and "Advanced". Below this is a section for "CHARTS" with "Area Chart", "Bar Chart", and "Bar List".

The main content area features a large heading "Why Another Chart Library" in bold white text. Below it is a question: "Why another charting library? There are some really good ones out there like...". A bulleted list follows, with each item in blue, suggesting they are links:

- Victory
- vx
- react-vis
- semiotic
- data-ui

At the bottom of the main content area, there is a explanatory text box with a light gray background and a thin black border. It contains the following text:

As we tried out various different frameworks, each of them were missing one thing or another that made them the PERFECT solution for us. Whether it was a missing chart type or the way it was animated and so on. Mixing and matching components from different frameworks to try and get that sweet spot will end up in a hodge-podge solution with things working differently.

**MDX**

**MDX** 💪

**MDX** 🤢

## Demo Dropdown Combobox

[Hide code](#)

```
<Dropdown.Combobox label="Demo Dropdown Combobox" value="" onChange={() => {}} options={[{  
    label: 'Banana'  
, {  
    label: 'Blueberry',  
    disabled: true  
}]} />
```

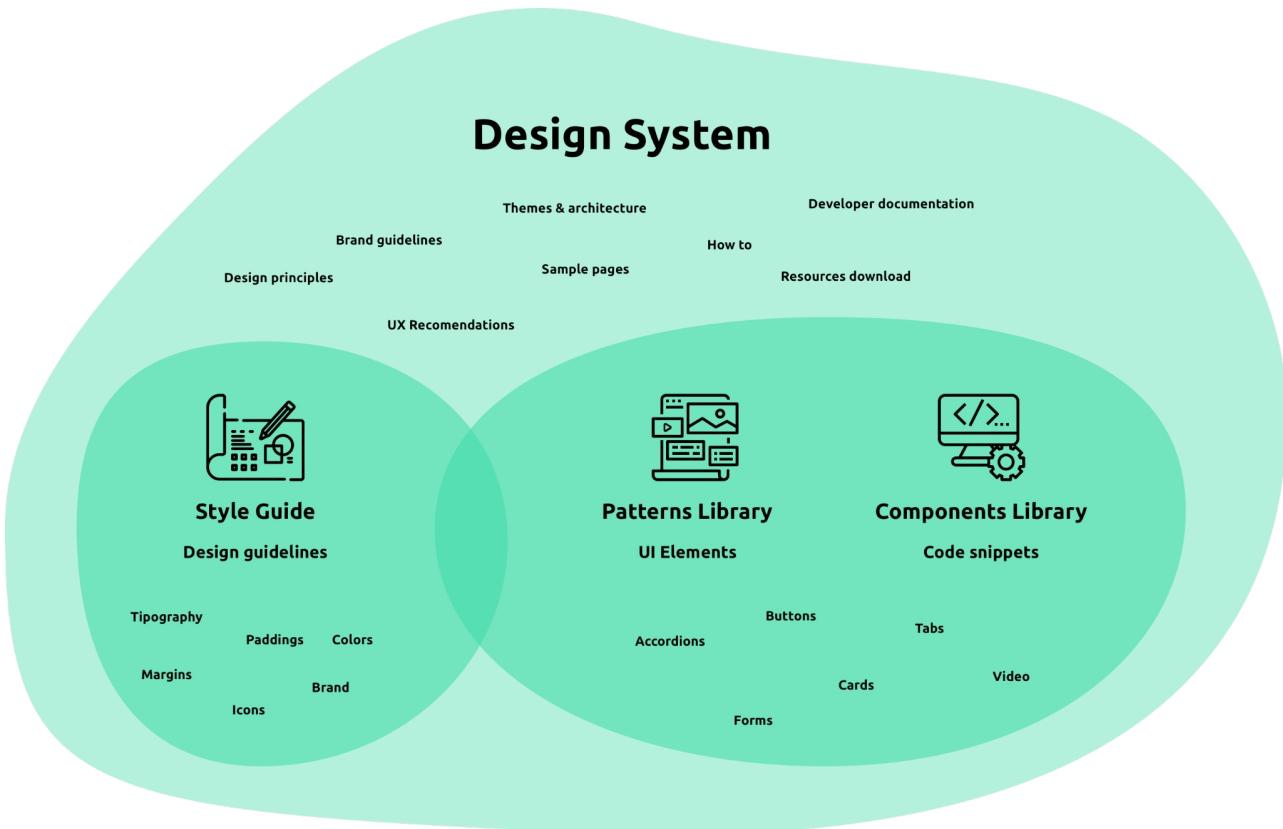
[Copy](#)

Case study

# Storybook for ~~design systems~~

component  
libraries

# Design System



<https://www.hi-interactive.com/blog/design-systems-from-a-ux-perspective>

The screenshot shows the Carbon Storybook interface. On the left, there's a sidebar with navigation links for 'GETTING STARTED' (Welcome), 'COMPONENTS' (Accordion, AspectRatio, Breadcrumb, Button, Danger, Ghost, IconButton, Secondary, SetOfButtons, Skeleton, Tertiary, Checkbox, ClassPrefix, CodeSnippet, ComboBox), and 'DOCS'. The 'Button' component is selected. The main area has tabs for 'Canvas' (selected) and 'Docs'. The 'Canvas' tab shows a blue button labeled 'Button'. Below it is a table titled 'Controls (18)' with columns for 'Name', 'Description', 'Default', and 'Control'. The first row shows 'as' with a description about rendering the button, a type annotation of 'func string elementType', and a control input field. The second row shows 'children' with a description about specifying content, a type annotation of 'node', and a control input field. The third row shows 'className' with a description about adding an optional className, a type annotation of 'string', and a control input field with a 'Set string' button.

Name	Description	Default	Control
as	Specify how the button itself should be rendered. Make sure to apply all props to the root node and render children appropriately <code>func string elementType</code>	-	-
children	Specify the content of your Button <code>node</code>	-	-
className	Specify an optional className to be added to your Button <code>string</code>	-	<button>Set string</button>

Carbon Storybook

The screenshot shows the Carbon Design System website interface. On the left, a sidebar contains navigation links: 'All about Carbon', 'Case studies', 'What's happening', 'Designing', 'Developing', 'Contributing', 'Guidelines', and 'Components'. Under 'Components', 'Overview', 'Accordion', 'Breadcrumb', and 'Button' are listed, with 'Button' being the active link, indicated by a blue underline. The main content area has a dark header with the word 'Button' in large white letters. Below the header is a navigation bar with tabs: 'Usage' (highlighted in blue), 'Style', 'Code', and 'Accessibility'. The main content area contains the following text: 'Buttons are used to initialize an action. Button labels express what action will occur when the user interacts with it.'

Carbon Design System

All about Carbon ▾

Case studies ▾

What's happening ▾

Designing ▾

Developing ▾

Contributing ▾

Guidelines ▾

Components ▾

Overview

Accordion

Breadcrumb

**Button**

Checkbox

Code snippet

Usage Style Code Accessibility

Buttons are used to initialize an action. Button labels express what action will occur when the user interacts with it.

The screenshot shows a web-based Storybook application for the Kaizen Component Library. The top navigation bar includes links for Home, Guidelines, Language, Components, and Storybook. The left sidebar features a navigation tree with sections for Introduction (selected) and Components (expanded), listing items like Accessibility, Avatar, Badge, Box, Brand, Brand Moment, Button, Card, Collapsible, and Date Picker. The main content area is titled "Welcome to the Kaizen Component Library". It contains two paragraphs explaining the purpose of Kaizen: providing standard components for efficient UI building while maintaining quality, accessibility, and consistency. Below this, a section titled "Getting started" is visible, along with a horizontal navigation bar at the bottom.

C Culture Amp

Home Guidelines Language Components Storybook

Storybook

Find components /

INTRODUCTION

Welcome

COMPONENTS

- Accessibility
- Avatar
- Badge
- Box
- Brand
- Brand Moment
- Button
- Card
- Collapsible
- Date Picker

Canvas Docs

Find components /

# Welcome to the Kaizen Component Library

Kaizen aims to provide standard components which enable teams to build UI as efficiently as possible while maintaining a high level of quality, accessibility and consistency with the overall platform.

To achieve this, Kaizen provides a set of React components and tokens which represent the foundations of any UI (colours, spacing, font sizes etc.). We aim to have what's represented here in code equally matching Figma, to enable cross-functional teams to communicate with the same language and ease of translation from design to code.

## Getting started

# **Do I need MDX?**

 Storybook | 2023

Sneak peek   Discord   Twitter   [Get your free ticket](#)

# Storybook Day

## March 14, 2023

10am - 12pm  
Pacific Standard Time

Watch live  
Online event

Brought to you by  chromatic

Join us for an online event about the future of UI development with Storybook. See what's new in 7.0, meet world-class frontend devs, and check out the leading projects in the community.

Your email address   [Get free ticket](#)

Free stickers inside



Storybook Day





```
$ npx storybook init
```