PatternFly Community Meeting

February 25, 2019

www.patternfly.org/v4d

Agenda

Topics

- New features and enhancements - Rachael Petrie, Michael Spaxman, Katie McFaul, Jessie Huff
- PatternFly 2020 roadmap - Dana Gutride
- PatternFly content enhancements: README files - Abi Donahue
- Navigation component visual design refactoring - Michael Celedonia
- Research on progressive form and wizard - Steve Cauffman
- A11y in PatternFly update - Jessie Huff

What's coming up

- What we're working on next - Rachael

Q&A

New features and enhancements

- Beta components on the website - Rachael Petrie
- Inline edit - Michael Spaxman
- Simple list - Katie McFaul
- Alert and Alert Group a11y- Jessie Huff

Topics

PatternFly 2020 Roadmap

Refining PatternFly efforts for 2020

Accessibility

- Audit all the components and address any regressions.
- Improve our test coverage.
- Scale a11y expertise across the community by conducting workshops.

Visual design

- Create RCUE (Red Hat Common User Experience) layer for Red Hat products to leverage on top of PatternFly to brand applications.
- Mature the visual design system by tightening up spacing and drop shadows.
- Address community feedback to improve component design.

Designer and developer experience

- Evaluate and revamp documentation on the website.
- Continue to build out starter projects.
- Integrate with Red Hat Developer tooling to provide quick starts for PatternFly.
- Keep the PatternFly Sketch library up to date.

Medium

PatternFly's Medium publication serves as another channel for keeping the PatternFly community connected and engaged. In January, we've seen an increase in followers, views, readership, and number of articles. Over the course of 2020, we'll continue focusing on featuring a wide range of topics written by all different writers, as well as developing the strategy and releasing articles on a regular basis.

Expanding PatternFly efforts for 2020

Components and enhancements

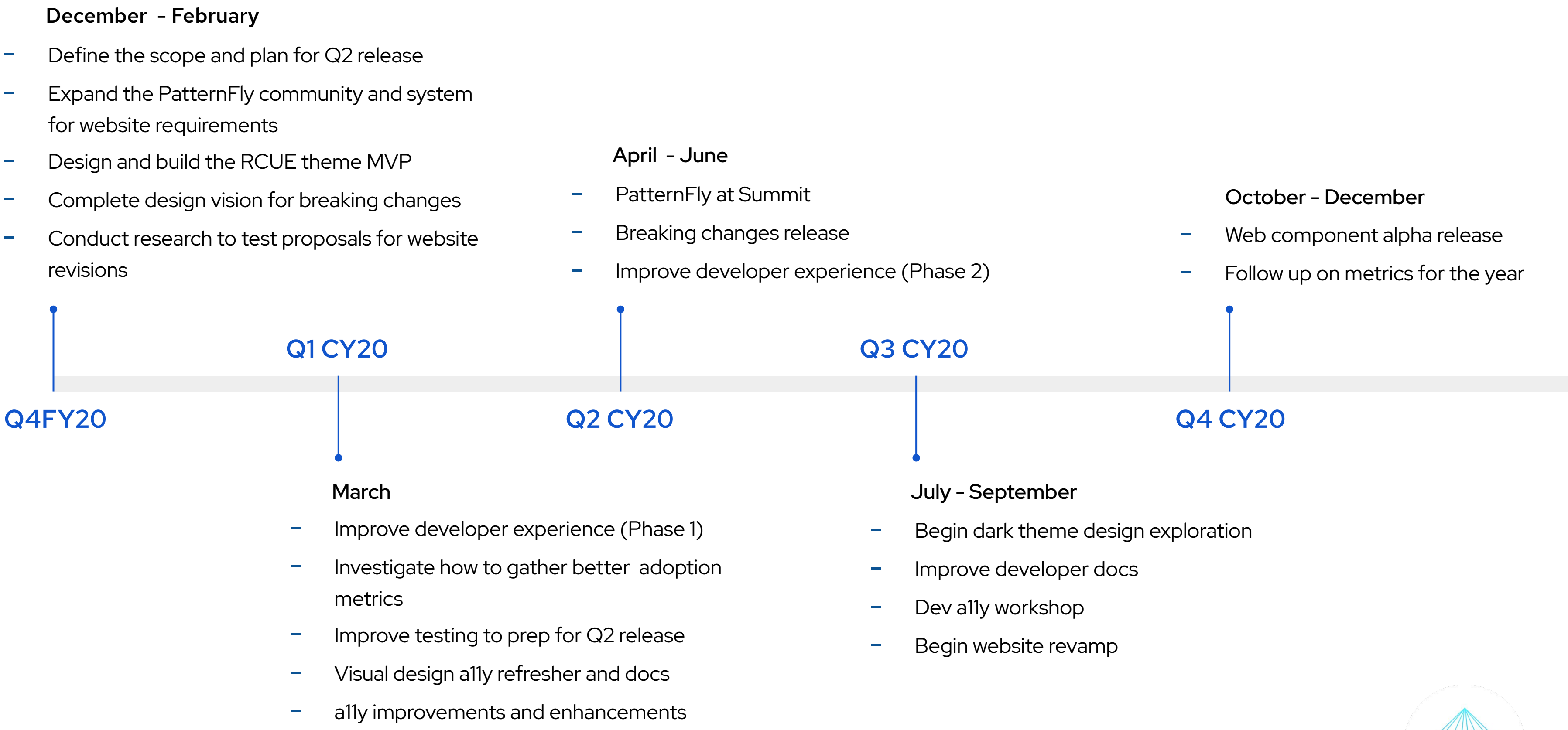
The library is more mature than ever, however there are still product demands for complex components and enhancements to the library.

Web components

This year, the team is committed to investing time for research and development in the area. The goal is to define a web component architecture that uses the existing PatternFly styles (CSS/HTML) and achieve complete alignment with the PatternFly design system. This engineering investment is future looking, and the PatternFly React components will continue to be the supported component reference architecture for PatternFly interaction.

Website use cases

PatternFly's focus has been to provide application UI for data-rich views and task workflows. To support a broader range of digital experiences, we'll expand PatternFly with components and documentation to meet website-specific or marketing use cases.



Q4FY20

December - February

- Define the scope and plan for Q2 release
- Expand the PatternFly community and system for website requirements
- Design and build the RCUE theme MVP
- Complete design vision for breaking changes
- Conduct research to test proposals for website revisions

Q1 CY20

March

- Improve developer experience (Phase 1)
- Investigate how to gather better adoption metrics
- Improve testing to prep for Q2 release
- Visual design a11y refresher and docs
- a11y improvements and enhancements

Q2 CY20

April - June

- PatternFly at Summit
- Breaking changes release
- Improve developer experience (Phase 2)

Q3 CY20

July - September

- Begin dark theme design exploration
- Improve developer docs
- Dev a11y workshop
- Begin website revamp

Q4 CY20

October - December

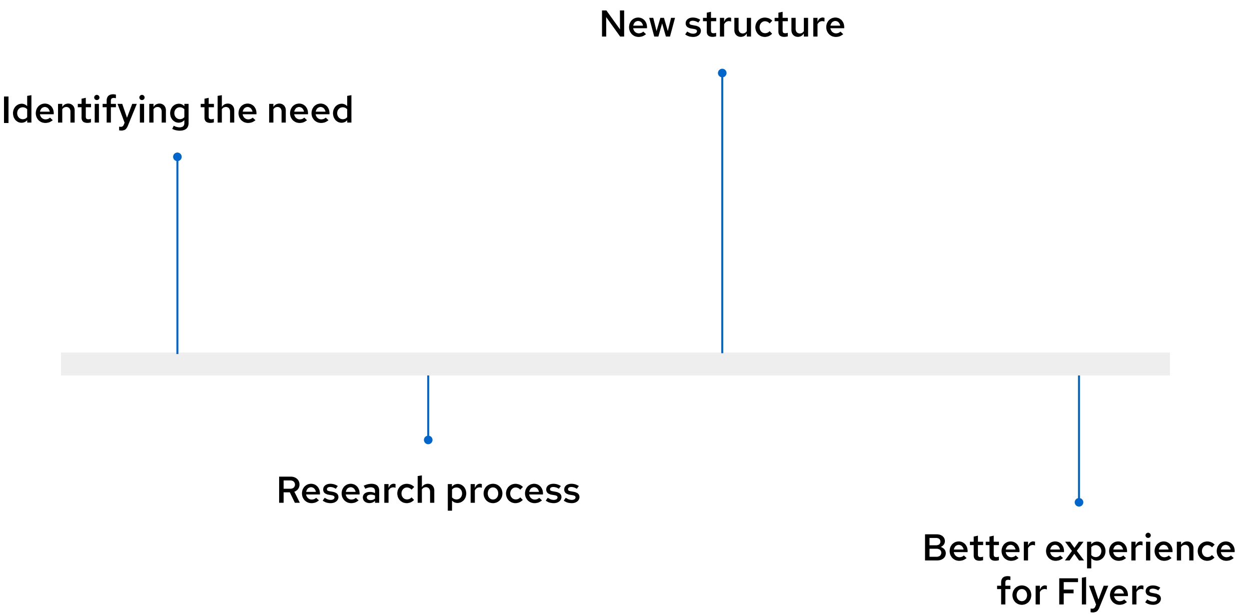
- Web component alpha release
- Follow up on metrics for the year

Q1 CY20 (March)

- **Revisiting metrics:** Investigate ways to crawl products to be able to show level of adoption.
- **RCUE:** As a follow up from last quarter's visual design exercise, implement the branding layer.
- **Web components and PatternFly core:** This quarter, PatternFly CSS experts will evaluate the web components POC and evaluate how to make use of PatternFly styles.
- **Infrastructure support for moving back to patternfly/patternfly:** The team would like to move core PatternFly back to that original patternfly/patternfly repo so we can officially drop the "patternfly-next" name.
- **Branch PatternFly and begin core changes for breaking changes:**
  - Variable updates and bug fixes
  - Tabs
  - Spacing
- **PatternFly developer experience (Phase 1)**
  - VS Code snippets
  - Che
  - OpenShift templates: Code ready workspaces (starter projects, possibly same as OpenShift templates)
- **a11y improvements and enhancements**
  - Visual design a11y refresher and docs
  - Complete all a11y issues identified in the audit.
  - Add a11y unit tests to react builds to validate dom structure
  - Investigate a11y tooling enhancements for axe or adopt IBM's tooling

PatternFly content enhancements: README files

Agenda



Identifying the need

Research process

New structure

Better experience for Flyers

Identifying the need

- Navigating the content
- Finding the content
- Comprehending the content

Research process

- Collaboration
- Discovery
- Learning

Research process

Collaboration

- Developers
- Designers

Discovery

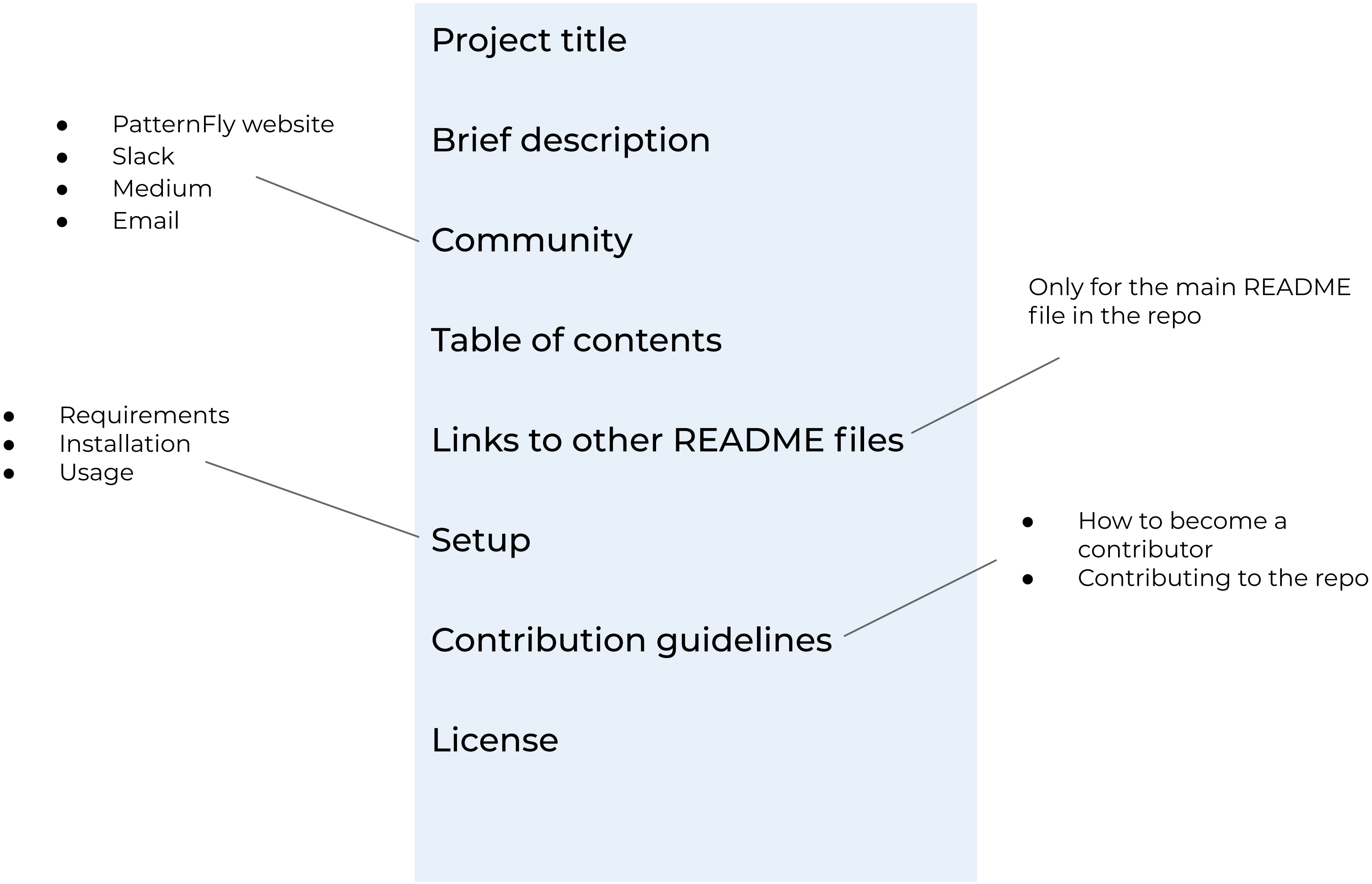
- Metrics
- Light audit

Learning

- How others do it
- Best practices

New structure

- Consistency
- Organization
- Readability



Project title

Brief description

Community

- PatternFly website
- Slack
- Medium
- Email

Table of contents

Links to other README files

Only for the main README file in the repo

Setup

- Requirements
- Installation
- Usage

Contribution guidelines

- How to become a contributor
- Contributing to the repo

License

Better experience for Flyers

- Finding information easily
- Navigating the documentation seamlessly
- Building product experiences

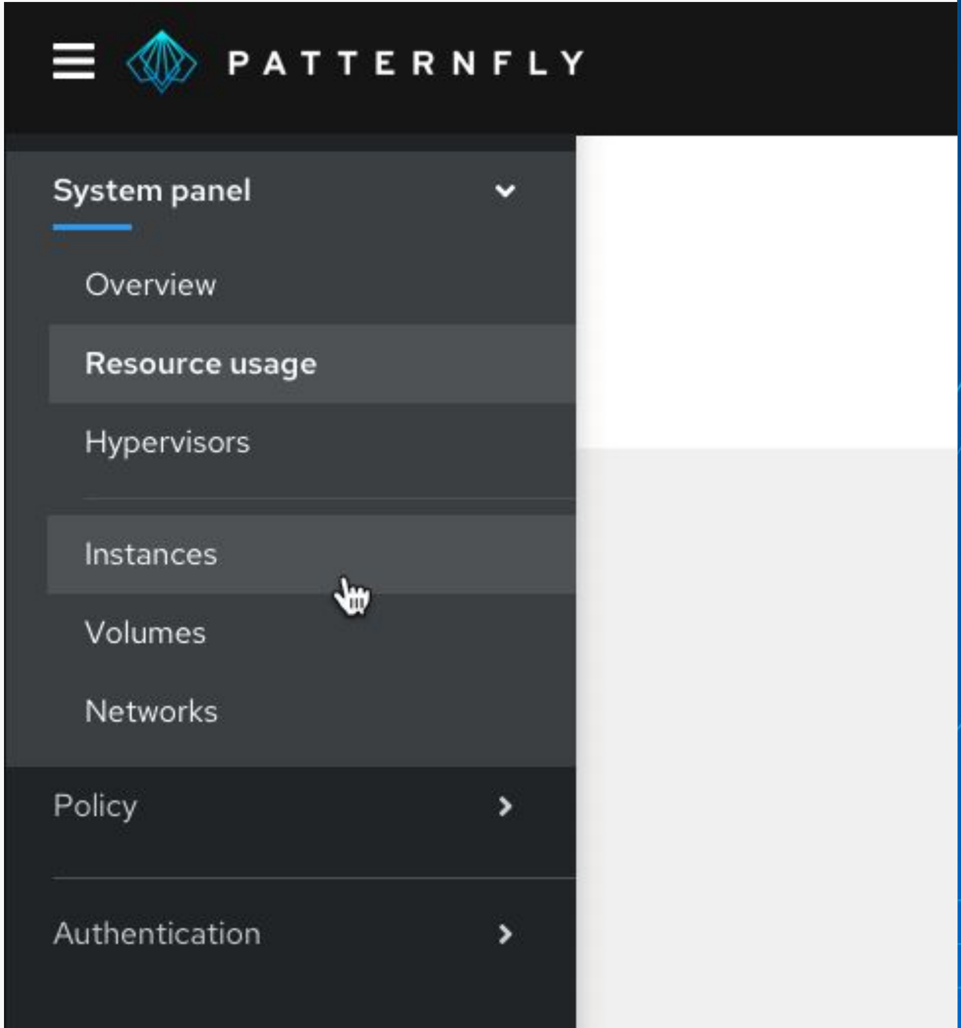
Navigation component visual design refactoring

Problem → Analysis → Solution

Problem

"...because the selected state provides little indication to the user about where they are in the nav."

Is this a symptom of a larger problem?



Analysis

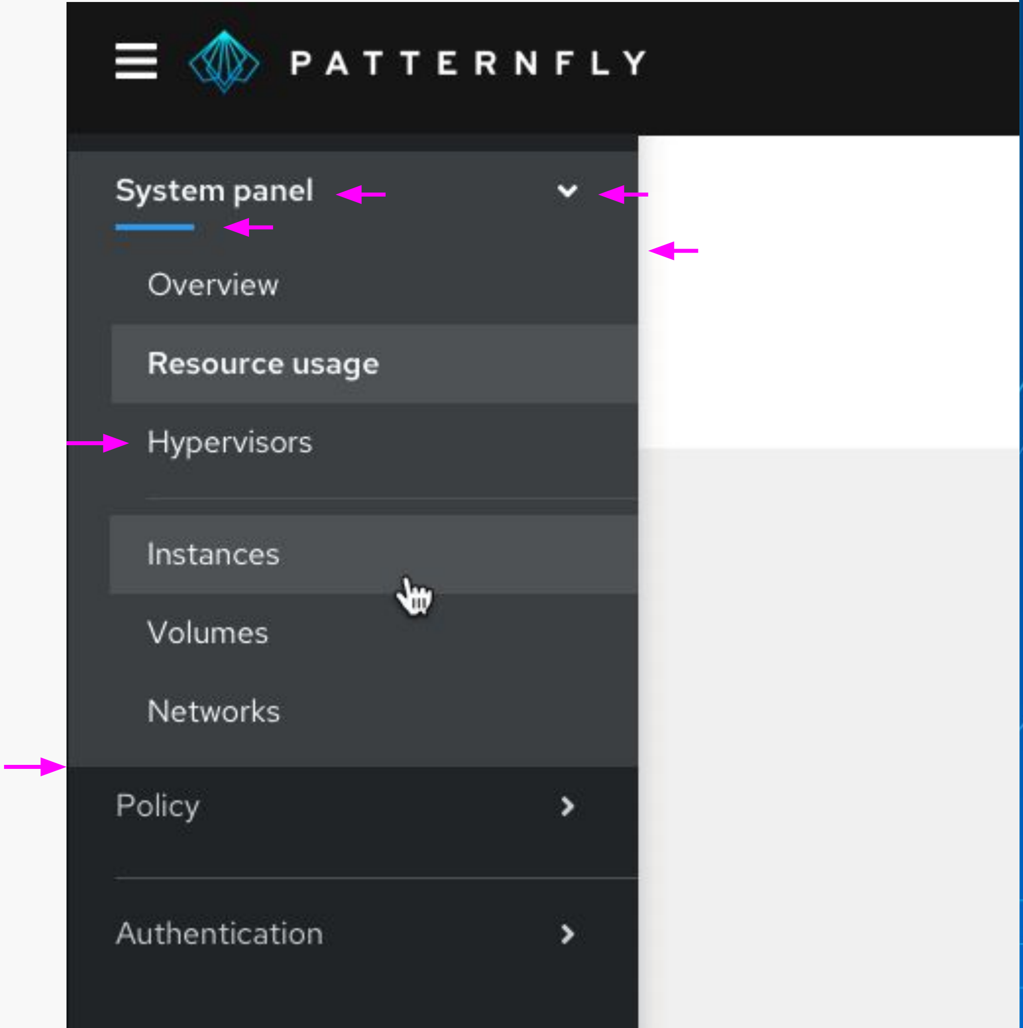
Too many visual indicators to tell us just two main ideas.

1. This is the page you are currently on.

- Background highlight
- Font weight change Reg > Med

2. This is an expanded top level item

- Background change
- Header font weight change Reg > Med
- Arrow icon points down
- Indented secondary items
- Blue bar underline



Solution

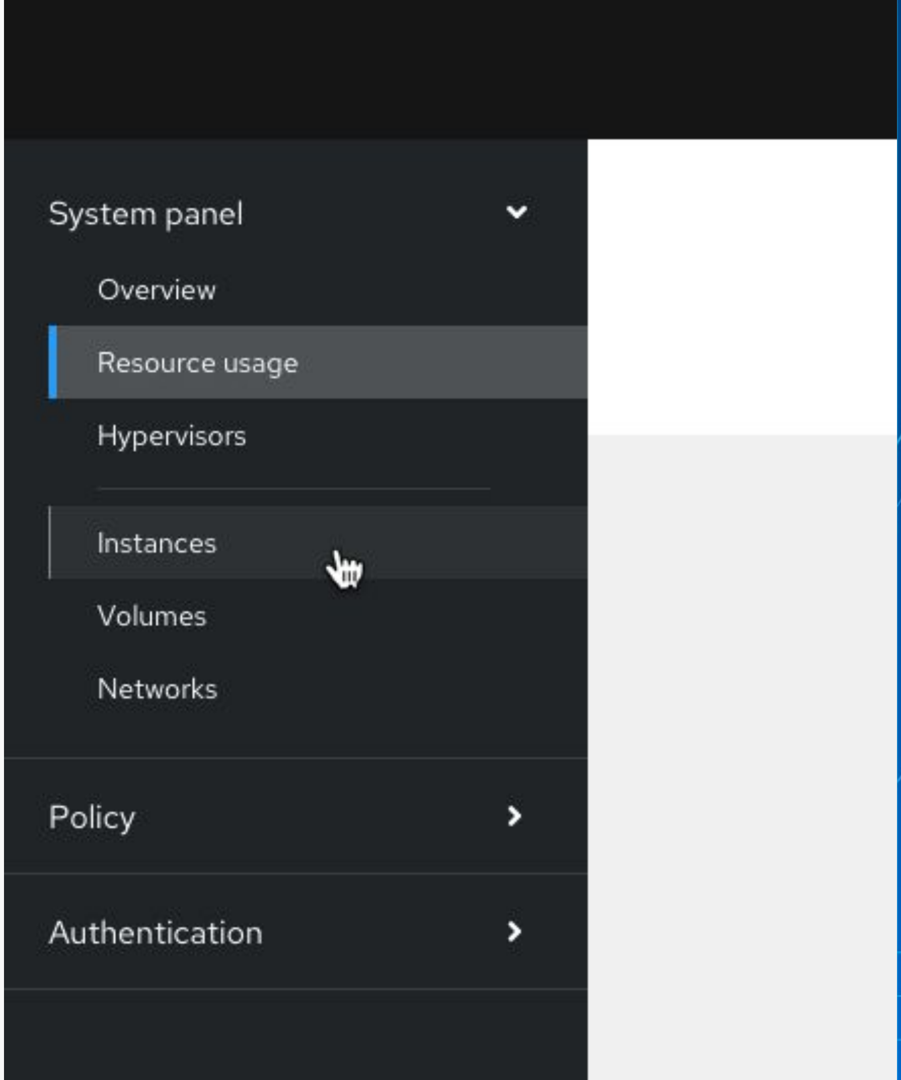
Refine visual indicators and refactor usage to highlight currently selected page.

This is the page you are currently on.

- Background highlight
- Blue vertical bar

This is an expanded top level item

- Font size change 16 > 14
- Arrow icon points down
- Indented secondary items



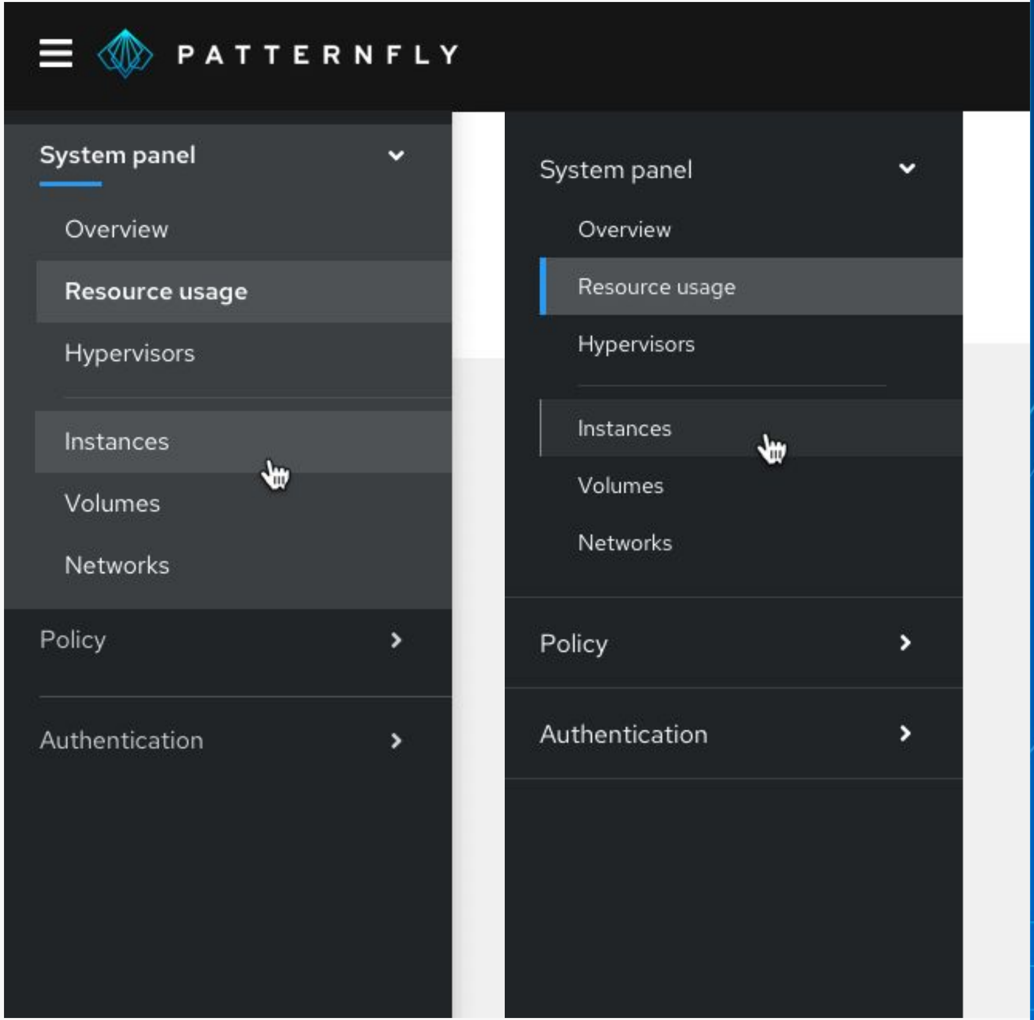
Solution

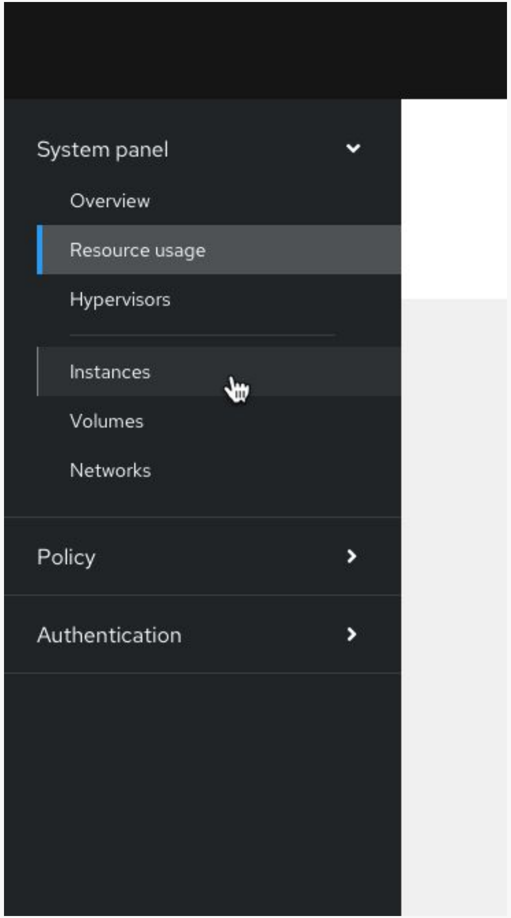
Visual emphasis is no longer diluted by unnecessary indications

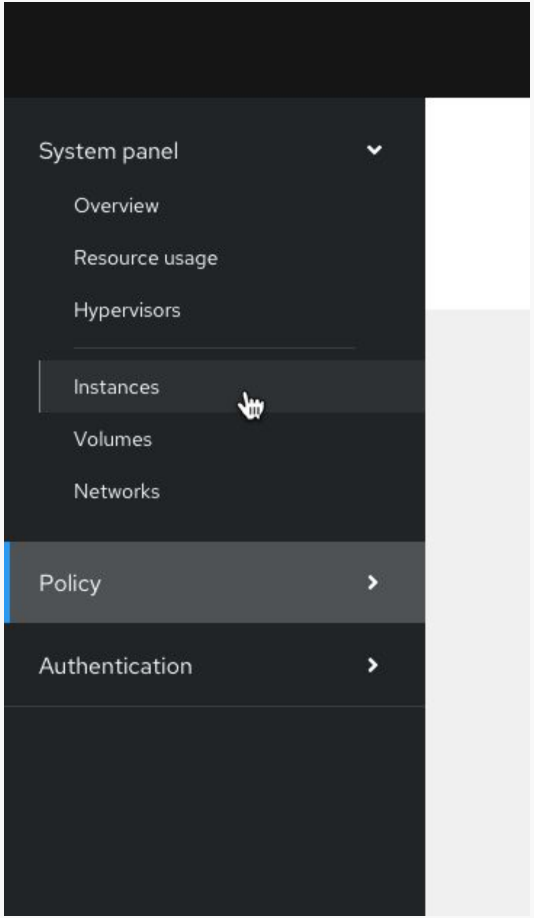
Blue bar used in a more appropriate fashion which scales to all other scenarios

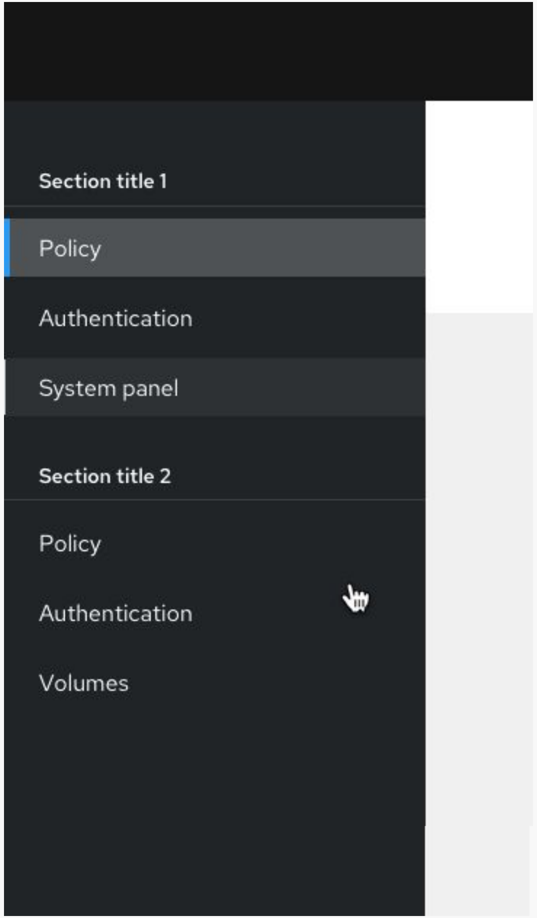
Clear communication of first and second level item hierarchy

Clear indication of currently selected item.




System panel
Overview
Resource usage
Hypervisors
Instances
Volumes
Networks
Policy
Authentication


System panel
Overview
Resource usage
Hypervisors
Instances
Volumes
Networks
Policy
Authentication


Section title 1
Policy
Authentication
System panel
Section title 2
Policy
Authentication
Volumes

Progressive Form vs. Wizard

An empirical comparison

Steve Cauffman

Pat Cox

Background

- The progressive form was initially advocated for by the Solution Engine team and within the OpenShift Dev Console team
  - Intended as an alternative to the Wizard for sequential task flows.
- The Wizard presents task steps one at a time to the user either in a modal dialog or within the page.
  - Requires the user to navigate by pressing “next” or “back” buttons.
- The Progressive Form presents task steps as a continuous scrollable page.
  - Allows for button navigation as well.
  - Scrolling means that user can review previous steps more fluidly.
- The PatternFly wanted to know if the Progressive Form was a component that should be prioritized for development

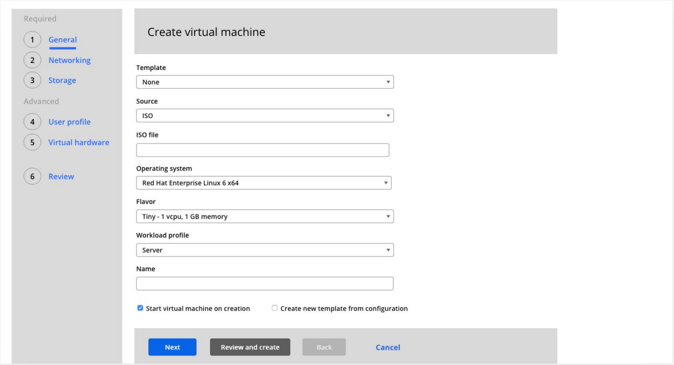
Current Study

- The current study sought to compare the progressive form and wizard and see if there were any differences in…
  - User performance
  - User preference
- Also wanted to determine…
  - a) If users wanted a progress stepper
  - b) What types of information should be in the stepper.

Method

- 7 participants were recruited from RDU List
  - Had a variety of backgrounds
- Mixed Methods Approach was used
  - Mixture of qualitative and quantitative measures.
    - Quantitative: Time on Task, Perceived Ease of Use, Perceived Efficiency
    - Qualitative: Preferences (Form Type, and Progress Stepper)
      - Elaboration on preference choice.
- Users performed 2 tasks within Progressive Form and Wizard
  - Order of forms was randomized
  - Task order within each form was not randomized
  - Allows for exploration of changes in performance when people are more familiar with task forms.

Mockups - Wizard



Create virtual machine

Template: None

Source: ISO

ISO file:

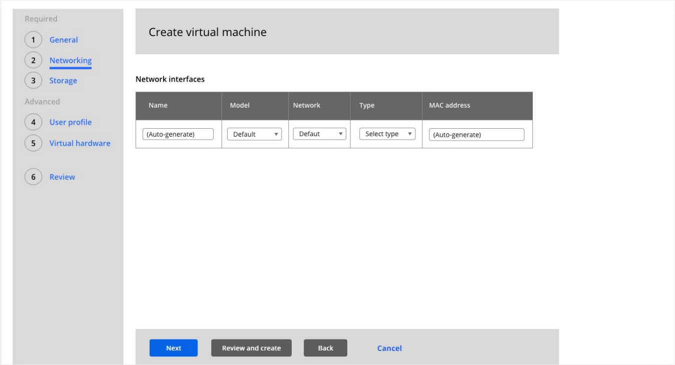
Operating system: Red Hat Enterprise Linux 6 x64

Flavor: Tiny - 1 vcpu, 1 GB memory

Workload profile: Server

Name:

☑ Start virtual machine on creation ☐ Create new template from configuration

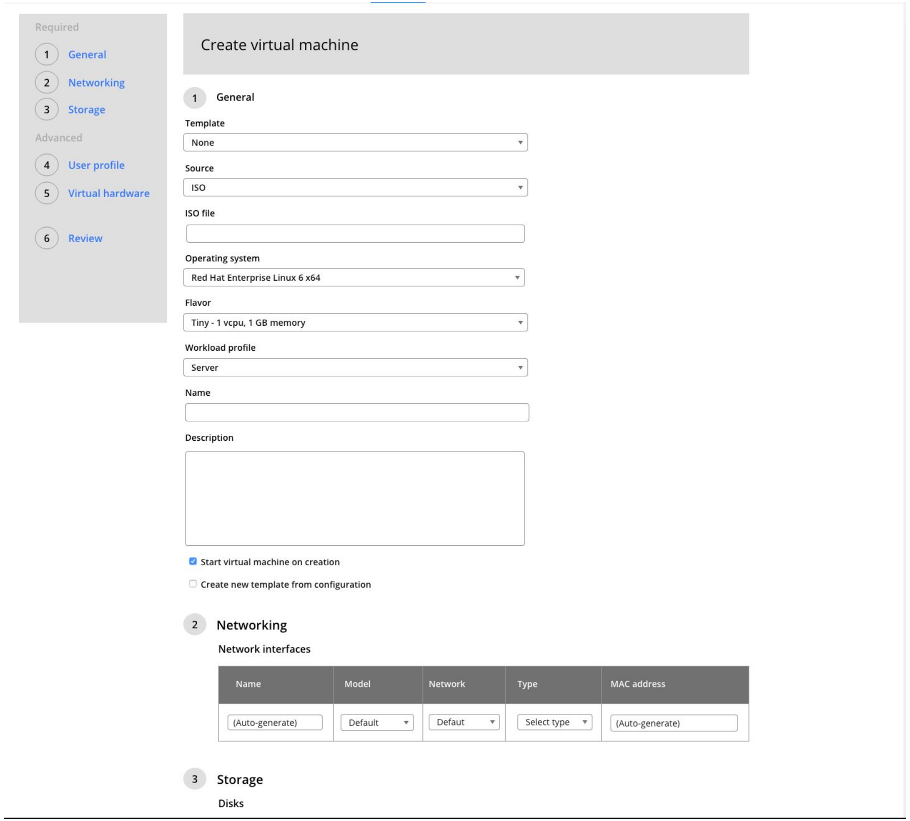


Create virtual machine

Network interfaces

Name	Model	Network	Type	MAC address
(Auto-generate)	Default	Defaut	Select type	(Auto-generate)

Mockups - Progressive Form



Required

1 General

2 Networking

3 Storage

Advanced

4 User profile

5 Virtual hardware

6 Review

Create virtual machine

1 General

Template: None

Source: ISO

ISO file:

Operating system: Red Hat Enterprise Linux 6 x64

Flavor: Tiny - 1 vcpu, 1 GB memory

Workload profile: Server

Name:

Description:

☑ Start virtual machine on creation

□ Create new template from configuration

2 Networking

Network interfaces

Name	Model	Network	Type	MAC address
(Auto-generate)	Default	Defaut	Select type	(Auto-generate)

3 Storage

Disks

Tasks

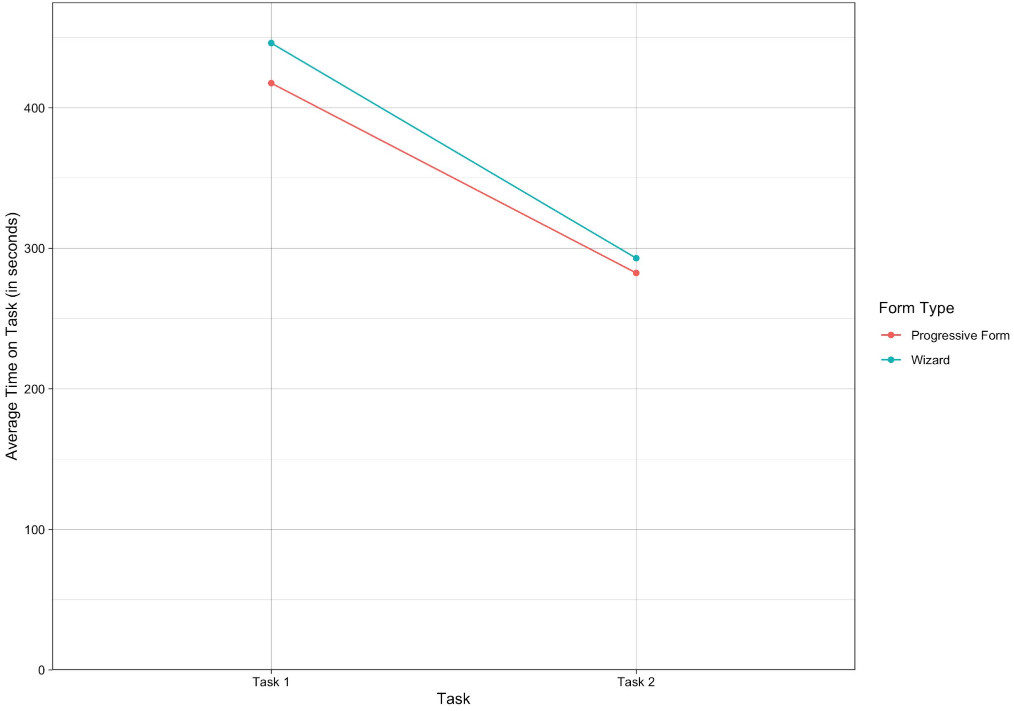
- Participants were asked to complete four tasks total.
- Tasks involved configuring Virtual Machines
  - Enough task information was given so that no experience was needed to complete tasks.
- Example Task 1:
  - Create a VM using a fresh RHEL install using the ISO file RHEL.iso and the following settings:
    - The version should be the most current version of RHEL, 16GB memory. Make sure it is configured to operate as a server with the name “TPS Reports”.
    - Set the network name to “Network interface 1,” the model to VirtIO, the Type to Local and the MAC address to 12.99.45.00.23.98
    - Create 5 disks at 10GB each with the source type as ISO and the interface type as VirtIO.
    - Create a user profile for “blumbergh” including a password of your choice.
    - Accept the defaults for other settings When you are done, check your settings
    - After reviewing the setting, you realize that you forgot to set the workload profile for ‘High performance’. Go back to change that setting and then when done, create the virtual machine.

Tasks

- Example Task 2
  - You want to create a VM template using a container without creating a new VM using the following configurations:
    - The source is called “lannister” and will use RHEL 7.
    - The template should generate a VM with a “small” flavor and be configured as a server. The name of template will be “baratheon”
    - The network device should be set to default settings with type set to “remote” with MAC address set to 45.65.76.22.67.91.
    - Make sure the storage source is set to ISO and add 3 disks with 20 gb each.
    - Create a user account for “nedstark” with password “jonsnow”
    - Review settings and create template.

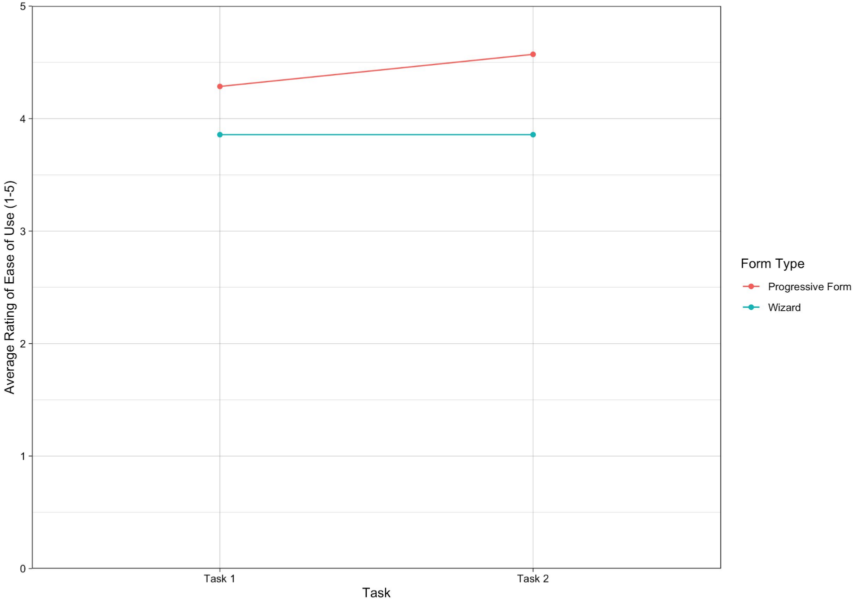
Results

Time Spent on Tasks



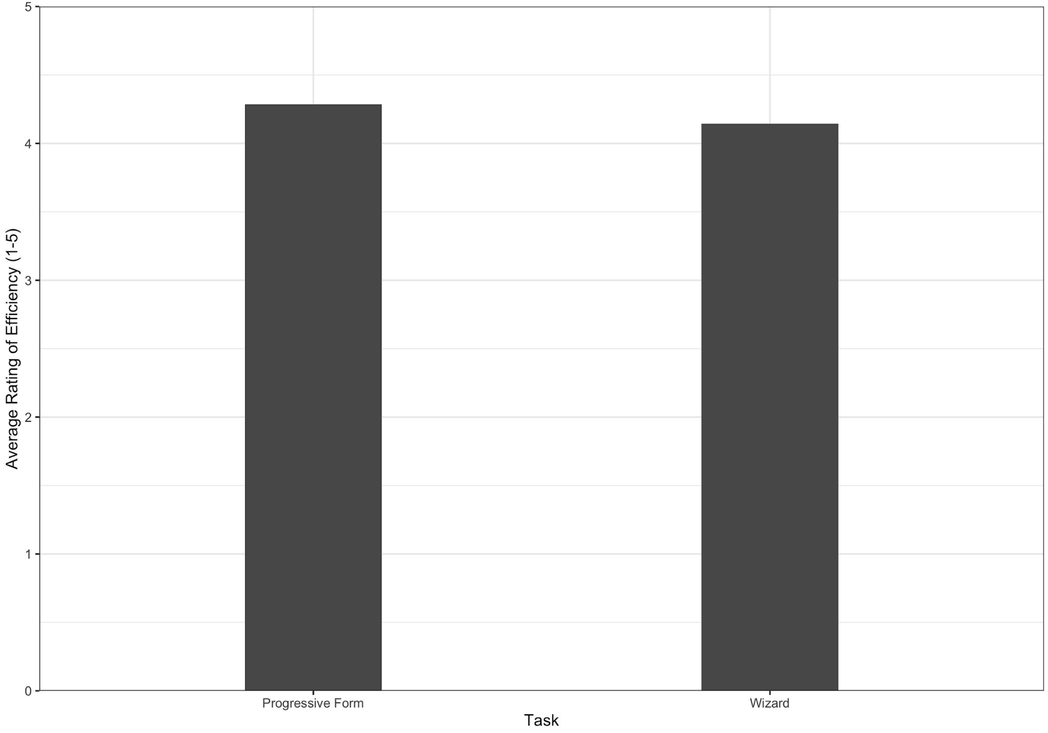
- Progressive form resulted in faster task completion times than wizard
  - This is a small difference though.
  - Difference also shrinks in the second task
- Both resulted in lower response times after users had some experience with the form.
- It’s possible that task 2 may have been easier in general for users due to less steps or less info input for fields.

Ease of Use



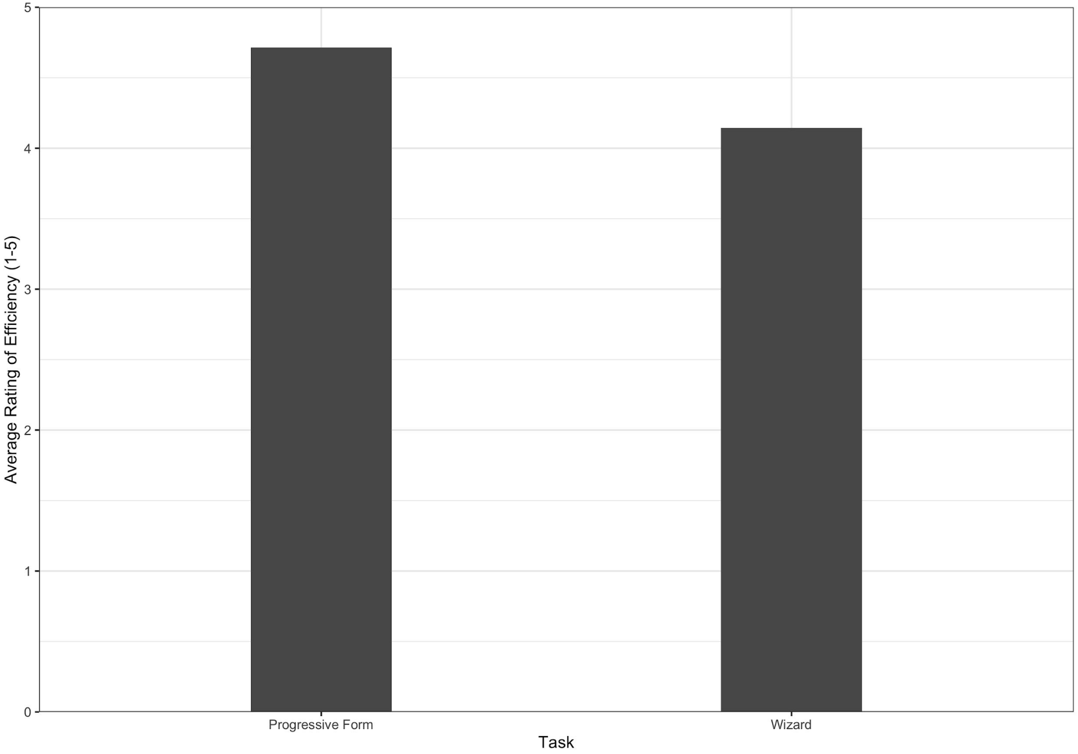
- Progressive Form is consistently rated higher in terms of perceived ease of use
  - Ratings increase as familiarity increases
- Wizard was rated at an average of 3.85 across both tasks.

Perceived Efficiency



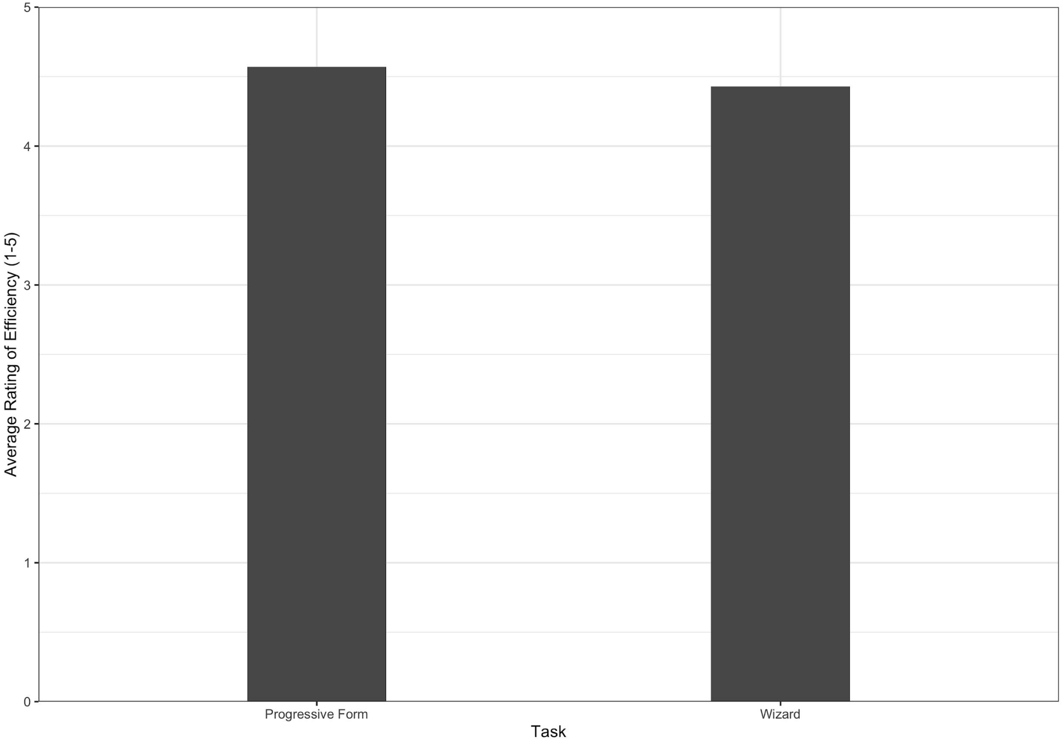
- Progressive form was rated slightly higher than the wizard in terms of perceived efficiency
  - This difference is marginal though
  - Statistically these are essentially equivalent.

Perceived Efficiency with Familiar Tasks



- Progressive Form was perceived to be more efficient for familiar tasks (i.e. tasks that users perform regularly in their day to day) than the Wizard.
  - This was also a small difference though (4.7 vs 4.1)
  - Most likely statistically equivalent.

Does The Form Meet the User’s Needs?



- Both forms received similar ratings when asked if they met the user’s needs.
- Progressive Form was rated higher but this difference is marginal

Which Prototype Do You Prefer for Sequential Task Flows?

- 6/7 participants selected the Progressive Form
- User Quotes:
    - “The progressive format with all choices on one single pane of glass makes it easier and more time friendly.”
    - “fewer clicks, I prefer to be able to see the "whole picture" and it's easier to correct things or check completed tasks if you're not sure you did them correctly”
    - “i see no reason for the input to be sectioned out into 3 pages. having everything on one page is more convenient, especially if a change has to be made earlier in the process. “
    - “Like the flow where everything is in one form, being able to scroll up to see all steps instead of exiting task to go back to another screen.”

Which prototype would you prefer for tasks that you are more experienced with?

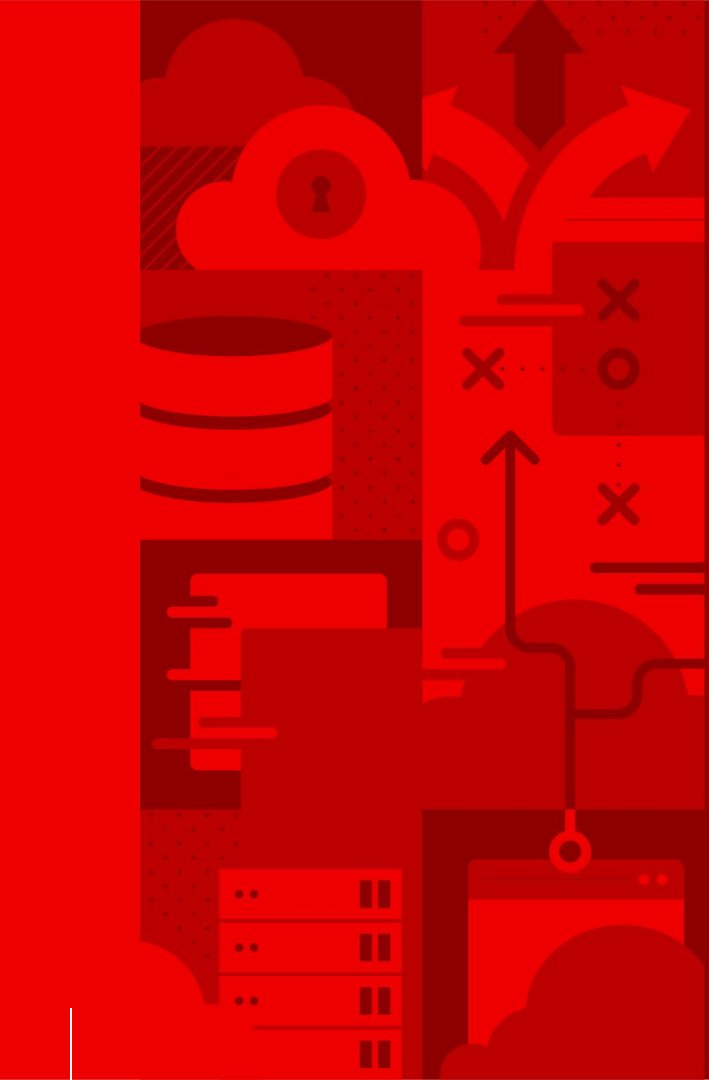
- 6/7 participants preferred the Progressive Form
- User Quotes:
  - “Just a simple, easy to use, quick form.”
  - “Clicking to the next screen takes extra time if i were trying to zip through these. it is also easier to review all choices when everything is on one screen and i don’t need to hit the back button to review.”
  - “Easier navigation is still the strong point, even with tasks that I'm more experienced with”

Which prototype would you prefer for task flows that require more than 10 steps?

- 5/7 participants preferred the Progressive Form
- User Quotes:
  - “again, I like to be able to see the whole picture”
  - “having the TOC on the left side is a good tool for a long form. i think there would be some limit to what would fit on one page, but 10 steps still seems like it would be OK on one page”
- Users who preferred the Wizard:
  - “To minimize cognitive load, I would like to be presented smaller chunks of related screens to complete a task with more than 10 steps.”
  - “The wizard lets me focus on the section that needs my attention. Too much information on the page is overwhelming”

Progress Stepper

- Participants were asked if they would prefer to have a progress stepper that shows their progression through a task flow.
- 5/7 participants responded that they would prefer to have a progress stepper.
- Also asked what kind of information they would prefer…
  - “For a task that I don’t do on a daily basis, I think a stepper would be useful. Usually a percentage to completion is the preferred label, like on a Kindle. It may help someone not give up on a task that feels tedious if they know how close to finishing they are.”
  - “If a section contains multiple sub-sections, then show me the step # and label. Otherwise, just the section header. “
  - “Step numbers, step labels, and a % Complete would be nice”



Summary

Summary

- The progressive form did show better performance (faster completion times) than the wizard when users were asked to complete a task flow in both forms.
- Progressive form was also rated as being easier to use, but there were marginal differences in terms of perceived efficiency between both forms.
- Users preferred the progressive form, in general.
    - The progressive form was preferred even for tasks that involve more than 10 steps; however, this is something that should be evaluated in a follow-up study.
- Users also would prefer to have a progress stepper with step labels, percentage complete, and step numbers.

Medium Article

- For more information

Accessibility Update

Determining where we are and where to go

Monthly audit of PatternFly:

- Axe testing: tests for clear violations of WCAG standards
- Keyboard interaction: can you get to it and interact with it by keyboard
  - Next step: consistency
- Screen Reader: Voice Over testing
  - Is it discoverable, understandable, usable
  - Structure
  - Content is clear and descriptive without being overwhelming

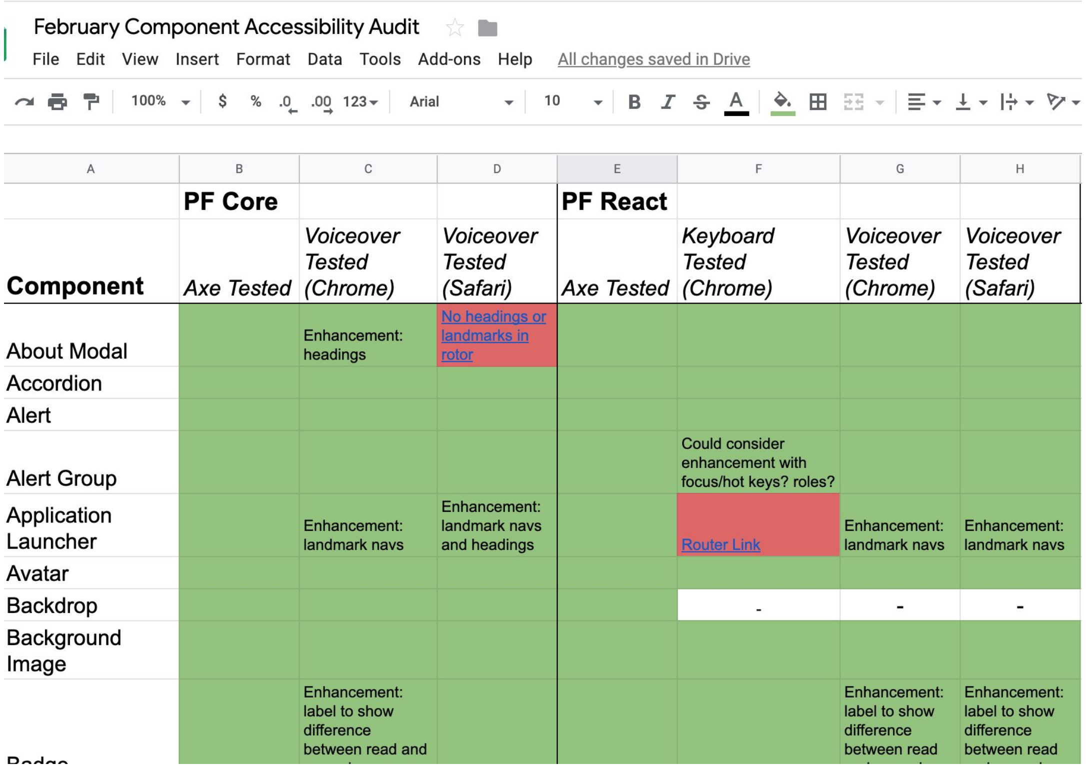
Note on the Audit

- Is it an accessibility issue with the **component**, the **examples**, or the **composition of the page** as a whole?
  - Difficult to always narrow that down and know right away
  - Just because we're seeing an accessibility issue doesn't mean it's always inaccessible to a consumer
    - (For example: keyboard accessibility of a component, perhaps should be fixed at component level or perhaps more appropriate to show within the example)
    - Area for improvement: better differentiating issues

PatternFly provides accessible components, but the next step is to build something accessible from those building blocks.

Accessibility Audit

https://docs.google.com/spreadsheets/d/1qWdqCn7iH_UW0RqupmgzZVauigRjccl7vI3Rqbq_i84/edit#gid=0


February Component Accessibility Audit
File Edit View Insert Format Data Tools Add-ons Help
All changes saved in Drive
100%
$ % .0 .00 123
Arial
10
A B C D E F G H

	PF Core			PF React			
Component	*Axe Tested*	*Voiceover Tested (Chrome)*	*Voiceover Tested (Safari)*	*Axe Tested*	*Keyboard Tested (Chrome)*	*Voiceover Tested (Chrome)*	*Voiceover Tested (Safari)*
About Modal		Enhancement: headings	No headings or landmarks in rotor				
Accordion							
Alert							
Alert Group					Could consider enhancement with focus/hot keys? roles?		
Application Launcher		Enhancement: landmark navs	Enhancement: landmark navs and headings		Router Link	Enhancement: landmark navs	Enhancement: landmark navs
Avatar							
Backdrop					-	-	-
Background Image							
Badge		Enhancement: label to show difference between read and				Enhancement: label to show difference between read	Enhancement: label to show difference between read

Findings from Audit

- **Axe:** a few component issues, mostly simple labeling
- **Keyboard interaction:**
  - a mixture of component and example issues, sometimes can be fixed in component other times should be shown in example
- **Screen Reader/VO:**
  - Structure and labeling
- **Summary:** overall in a good position accessibility wise, a few areas that have slipped through the cracks potentially from regressions and areas for improvement

Priorities

#1 Priority: Accessibility in Our Components– if it's in our control, it should be accessible

#2 Priority: Accessible Examples– showing consumers the accessible way to handle the use case is extremely helpful and increases the likelihood of proper accessibility in products

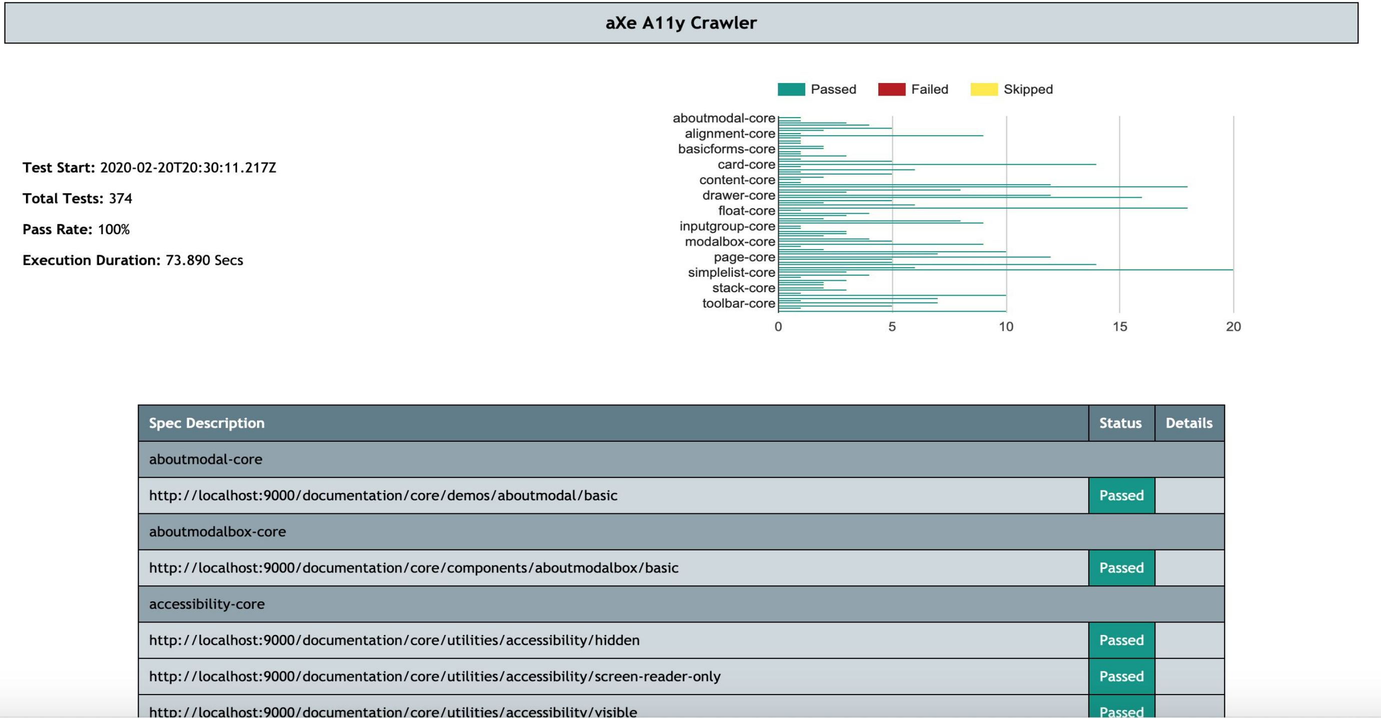
★ Prioritizing issues based on accessibility issues with components (further prioritized by product need and usage) and then examples when identified

How have we addressed these findings?

- **Axe**: added ability to automate this testing
    - Script that runs axe against components monthly
    - Added axe validation check to the CI Build so every PR is checked through axe
- **Keyboard**: issues created and prioritized, ensured for new PR additions
    - Area of improvement: regressions and testing
- **Screen Reader**: improved structure of all of our pages, evaluated current labeling, took on specific component issues

Automated Accessibility Testing

https://4475-122507459-gh.circle-artifacts.com/0/coverage/index.html
https://27680-84372347-gh.circle-artifacts.com/0/packages/patternfly-4/react-docs/coverage/report.html



aXe A11y Crawler

Test Start: 2020-02-20T20:30:11.217Z

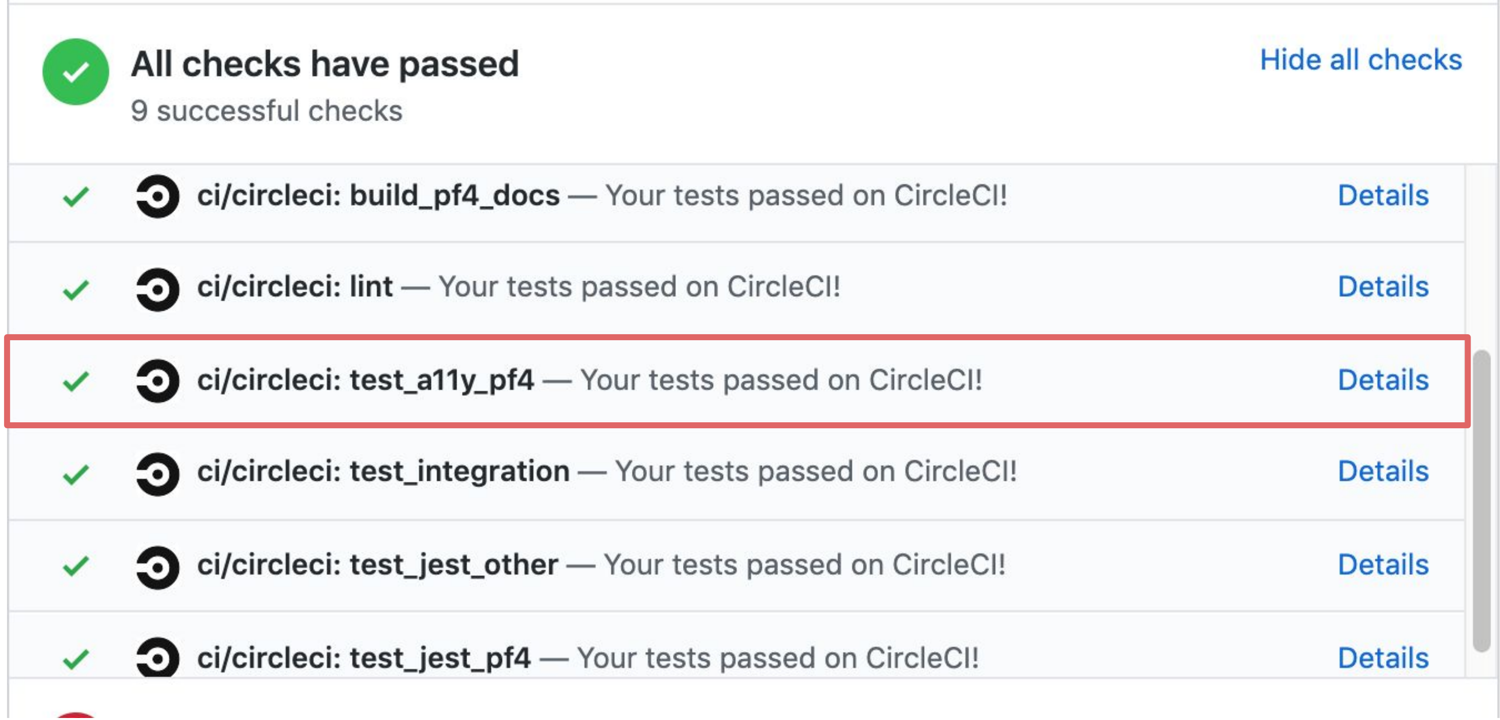
Total Tests: 374

Pass Rate: 100%

Execution Duration: 73.890 Secs

Spec Description	Status	Details
aboutmodal-core		
http://localhost:9000/documentation/core/demos/aboutmodal/basic	Passed	
aboutmodalbox-core		
http://localhost:9000/documentation/core/components/aboutmodalbox/basic	Passed	
accessibility-core		
http://localhost:9000/documentation/core/utilities/accessibility/hidden	Passed	
http://localhost:9000/documentation/core/utilities/accessibility/screen-reader-only	Passed	
http://localhost:9000/documentation/core/utilities/accessibility/visible	Passed	

Axe Testing on Build



✓ **All checks have passed**

9 successful checks

✓ **ci/circleci: build_pf4_docs** — Your tests passed on CircleCI! Details

✓ **ci/circleci: lint** — Your tests passed on CircleCI! Details

✓ **ci/circleci: test_a11y_pf4** — Your tests passed on CircleCI! Details

✓ **ci/circleci: test_integration** — Your tests passed on CircleCI! Details

✓ **ci/circleci: test_jest_other** — Your tests passed on CircleCI! Details

✓ **ci/circleci: test_jest_pf4** — Your tests passed on CircleCI! Details

Issue Progression

- **Keyboard Related Issues:**
  - Dropdown keyboard interaction and tests,
  - Application Launcher- new favorites feature
  - Documentation Updates- Skip to Content

- **Screen Reader Related Issues:**
  - Table accessibility in VO
  - Custom page labeling for VO
  - Accessible roles in Page and PageHeader
  - Alert accessibility updates and Alert Group
  - Documentation Updates

What’s coming up

What we’re working on next

- Enhancements
  - Compact data list (for show/hide modal)
    - Link to design
  - Model enhancements
    - Link to list of modal enhancements
  - Adding filtering to the checkbox select
    - Link to design
- Features
  - File upload
    - Link to design

- See our current milestone in progress here.

Q&A