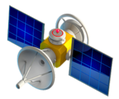
COMMON COMPONENTS

Sprint Review

Troposphere 

Agenda

01 Presenting our user research findings

02 Launchpad Demo - Learnings from GitHub Actions

03 Lessons Learned, Help and Feedback

User Research

User Research

Team Interviews

Invasive Species

Common Services Showcase

BC Gov Design System

Registries (Relationships)

Health Gateway

BC Developers Kit

Real Property Division PIMS

Mines Digital Services

User Research

Team Interviews - What did we learn?

Inception

Proof of Concept

Minimum
Viable Product

Active Product
Development

User Research

😀 Interactions 😕

- Reliance on Scrum Master
- RocketChat, OpenShift, IDIR
- Business area onboarding
- Teams are familiar with Agile
- Limited guidance on choices for tech stack
- Limited awareness of what other teams are using

💪 Opportunities

- Resources to build awareness around technologies used at the lab
- Target new teams first
- Example applications that use OpenShift
- Backends optimized for GIS

🙋 Internal Ownership

- Follow team journeys and coco needs
- Join first sprint review, ensure teams feel supported
- Help shape coco adoption from onset

Inception | Proof of Concept | Minimum Viable Product | Active Product Development

User Research

😀 Interactions 😕

- Teams take ownership, make stack decisions independently
- Google search the primary place to look up components
- DevOps is the first hurdle to overcome, sometimes with a lack of guidance.
- Delay in OpenShift training may slow down momentum
- No unified style guide as a baseline for front-end work
- Gov systems that are slow with on boarding could lead to friction (e.g. Jira, IDIR)

💪 Opportunities

- Understand teams tend to look outward first
- Provide recipes rather than policies around best practices
- Resources that may help teams with open shift deployments prior to training
- Ensure resources provided also include when NOT to use a coco

🙋‍♀️ Internal Ownership

- Follow team journeys and coco needs
- Join first sprint review, ensure teams feel supported
- Help shape coco adoption from onset

User Research

😀 Interactions 😕

- Base infrastructure is all in place
- Feature development in full swing
- Solutions need to be found for baseline features to be completed before business goals are met
- Early choices now have a greater impact. Teams may need to course correct
- DevOps begins trending towards maturity, but perhaps still a struggle

💪 Opportunities

- Clear product roadmaps may indicate a need for potential cocos
- Begin looking into whether service based solutions for teams may be a good fit

🤷‍♀️ Internal Ownership

- Provide a level of service that makes team look forward to using cocos
- Clear communication on how and when to use our services

Inception

Proof of Concept

Minimum Viable Product

Active Product Development

User Research

😀 Interactions 😕

- Actively sprinting and deep into feature dev to hit goals
- Ingrained in lab culture, likely to carry on post graduation
- Component needs may pivot towards enhancing the user experience
- Tech debt becomes a higher priority

💪 Opportunities

- Recognize feature gaps that more mature projects have when compared to solutions that exist for private citizens
- Check in on current state of the team and their needs. Is there a need that has arisen that other teams may benefit from?

🙋‍♀️ Internal Ownership

- Research feature gaps that exist for mature products
- Check in on adoption and continuous improvements

User Research

How Teams Evaluate Components

Easy to Find

Teams want a simple place to go to find common components. This should have great SEO, as they often go to Google first.

Usage

Common components aren't common if they aren't being used. Teams like to see usage data, like the number of services onboarded or the number of transactions.

Support

Teams hate shelfware, and want to know that if they onboard to a service, someone is maintaining it. Teams running critical apps also care deeply about uptime and SLAs.

Simple Descriptions

Teams need simple, plain-language descriptions of common components, including key features. This should be readable by non-technical people as well.

Reviews

Once they know a component is being used, the next question is always whether people like it. In absence of reviews, teams use the BC Gov architect shadow network.

Easy onboarding

Once they've evaluated this information, teams look for developer documentation with clear instructions and sample implementations to make onboarding easy.

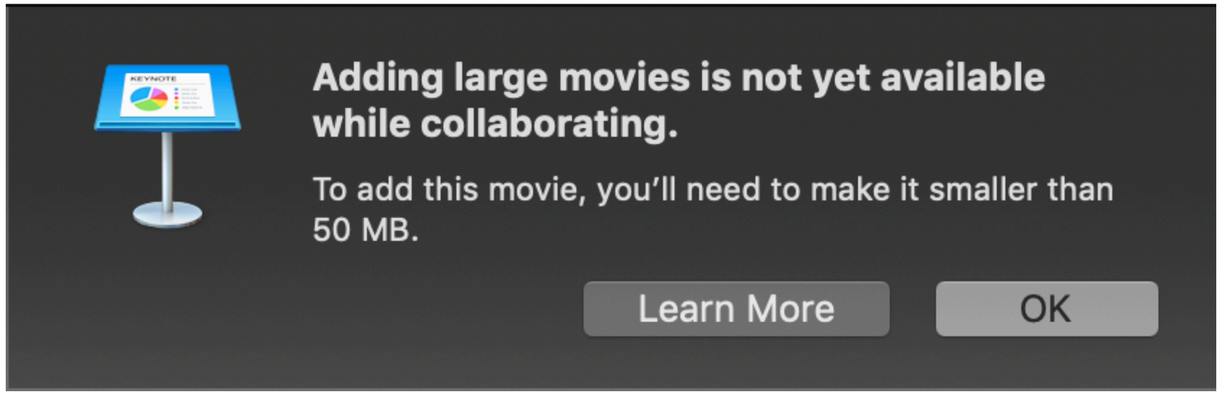
GET DEVELOPMENT STARTED AT

LIGHT Speed

.com/bcgov/openshift-launchpad

Launchpad

GitHub Actions Video

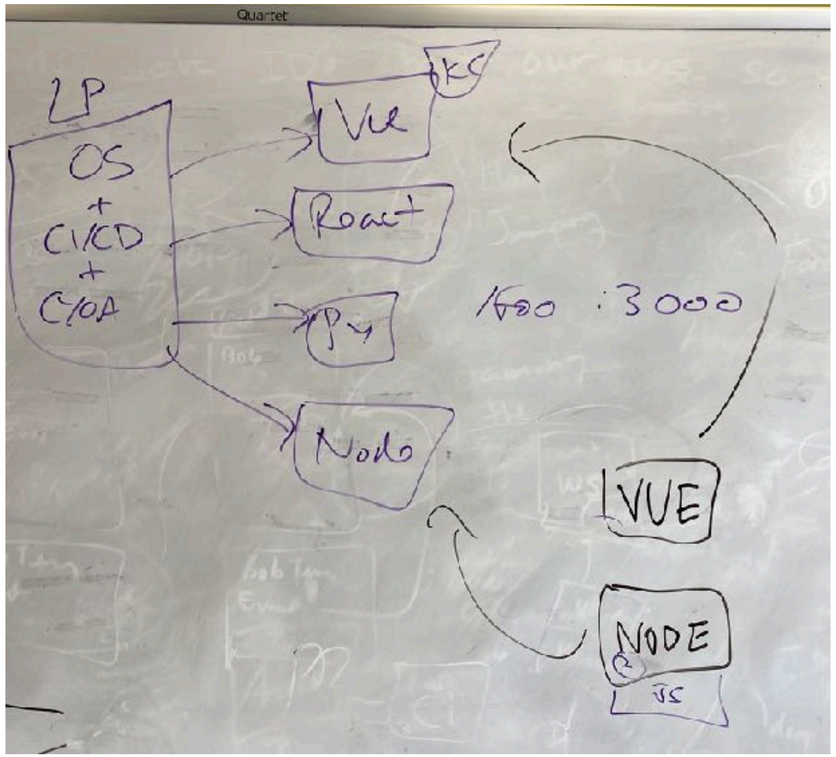


Click here to go to the Jira ticket that goes to the video

Launchpad

Our First Contribution!

Shoutout to the Common Services Showcase Team for contributing Vue and Node!



Launchpad

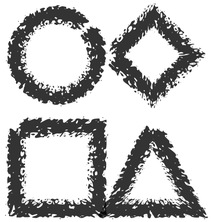
Our Product Poster!

Lessons-Help-Feedback

Lessons-Help-Feedback

Lessons We Are Learning

Shaping (in theory)

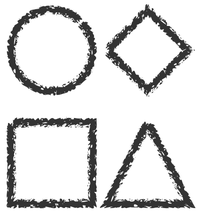


Shaping (reality!)

Lessons-Help-Feedback

Lessons We Are Learning

Shaping (in theory)



Shaping (reality!)

Shaping means coming to the team with problems and figuring out solutions TOGETHER

Lessons-Help-Feedback

We’d love your feedback

Does our User Journey match your experience?

Would you get value from our GitHub Actions example? Why or why not?

If we continue working on Launchpad, what would you like to see next?

Lessons-Help-Feedback

We could use your help

Please contribute your thoughts on Launchpad. We will have a public PR available for comment.

Do you have a different option you'd like to add to Launchpad? We'd love to collaborate!

We need to submit our PIA & STRA. Any resources we should use?

Lessons-Help-Feedback

Contact Us!

#team-coco

#common-components

Exchange Lab

(3rd Floor)

jfournier@freshworks.io

kendall@freshworks.io

rachit@freshworks.io

Thank You

From #team-coco

BRITISH COLUMBIA

freshworks STUDIO

CGI