Sprint 6 Retro Meeting

1. **Reflection**
    a. Sprint
        i. What went well?
        ii. What could have gone better?
    b. Project
        i. What went well?
        ii. What could have gone better?
2. **Demo**
    a. Playbook
    b. Solution
3. **Conclusion**
    a. Final thoughts
    b. BC Exec Presentation

Sprint

What Went Well?	What Could Have Gone Better?
● Solid conclusion ● Robert helping with documentation ● Everything ● Managing other commitments ● Commitment from all involved	● Managing Camosun shifting requirements ● Tight one week sprint, extra pressure ● Heavy load of incoming email

Project

What Went Well?	What Could Have Gone Better?
● Team dynamic ● Communication ● Keeping on track ● Supportive team ● Autonomous and interdependent team	● Challenges with Covid ● Students lacking a framework in the beginning ● More client engagement ● More planning up front ● More thought put into architecture

Sprint 5 Retro Meeting

1. **Reflection**
    a. What went well?
    b. What could have gone better?
2. **Demo**
    a. Solution Updated UI
    b. Solution PWA Update Notification
3. **Final (Half) Sprint**
    a. Solution Documentation
        i. Fill out Robert's Template
    b. Comment and Clean Up Code
        i. SoC and DRY
    c. 12 Factor Documentation
        i. Workshop Meeting
    d. Playbook Dev Guide Walkthrough
    e. Solution Dev Walkthrough as well
    f. BC Exec Presentation (Tuesday, August 4th at 11:30AM)
        i. What do we want to show here?
        ii. "30 Minutes and 4 slides"
            1. What did we do
            2. This is what we learned
            3. How would we have done things differently
            4. Demo of both (emphasis on solution maybe)
                a. Why is the solution relevant, frame around starter kit
        iii. We have class! Need to reschedule...
    g. Half sprint - sprint retro on Friday, July 31st?
        i. Let's see how the week goes...

What Went Well?	**What Could Have Gone Better?**
● Responded to ongoing feedback throughout the sprint (handling change requests in a timely manner) ● Commitment to solving problems ● We continued to learn, and apply our learning in different ways ● The team accepting change	● Could have had more structure around the tasks could have reduced the amount of time they took ● Still thinking about how more planning at the beginning of the project could have helped ● Never nailed down the "how" of what we were going to do

Sprint 4 Retro Meeting

1. **Normal End of Sprint Retro**
    a. What went well?
    b. What could have gone better?
2. **Demo**
    a. Playbook Beta
    b. Solution Cookies
    c. Solution PWA
3. **Possible Next Sprint Goals**
    a. Solution
        i. Implement Redux
            1. Probably not
            2. Instead, let's create documentation that explains how the solution is event-driven
                a. Be able to tell a story about why the solution ended up the way it is, and include some elements of architecture in there
                b. Images and words
        ii. UI refinement
        iii. Cookie Notification and Acceptance (likely no 1 priority)
    b. OpenShift
        i. Telemetry* (logging)
    c. Playbook
        i. Beta Release
        ii. Take Feedback
        iii. Maybe start planning another release
        iv. Look into programmatic version tag
        v. Start planning transition
4. **Look at Remaining Sprints Document**
    a. Discussion on the rest of the project

What Went Well?	**What Could Have Gone Better?**
● Nice and productive (x2) ● Murphy pivoted around his task to do the PWA ● Josh kept on working, even though times were tough ● Kelly and Robert collaboration ● Productivity based on self-learning and implementation	● People working in silos ● Could have more frequent pushes to the personal GitHub repos ● Looking back, could have spent more time nailing down MVP

● Going down a rat's nest and coming out clean on the other end ● Clean code ● Trello use ● Good team synergy	

Sprint 3 Retro Meeting

1. **Normal End of Sprint Retro**
    a. What went well?
    b. What could have gone better?
    c. Any issues?
2. **Demo**
    a. Playbook Issues and Development Guide
    b. Solution Pipeline
    c. Solution Architecture
3. **Next Sprint**
    a. Possible goals for next sprint
        i. Database for solution
        ii. Database for OpenShift
        iii. Database Documentation
        iv. Playbook issues as they come up
            1. Playbook header and footer
            2. Revised plays
        v. Playbook grammar first pass?
        vi. Preparing for playbook beta
    b. Trello board review
        i. Notes on Trello cards

What Went Well?	**What Could Have Gone Better?**
● Use of "Issues" on the playbook github ● A lot of progress on foundational work (pipeline, playbook feedback, updated codebase on github) ● Communication ● Teamwork - Putting our heads together and figuring out GitHub issues	● Too much pressure on Murphy ● Trello Cards ● Expectations on output

Sprint 3 Retro Notes

VERY IMPORTANT:

- We need to write our process we went through solving out GitHub issues
- What was the problem? How did we solve it? Any warnings for future groups?
- Need to note why we changed folder structure (redundancy)

Playbook:

- Playbook is actually being used by BC Gov already!
- Robert wants a build guide, Kelly already made one

Pipeline:

- Pipeline API testing is great

Solution:

- We need to create better Trello cards to define our goals better
- Robert wants us to take a second look at our architecture
- Before we build a database redo Trello cards
- Grant's thoughts: Instead of creating a database use cookies so we can save time
- We should take a couple days to think it over
- Grant doesn't like a login feature
- We need our user stories to look something like this:
    - As a user I would like to tag stories so I can access them easily at a later date
- Robert wants us to look into an installable progressive web app

Sprint 2 Retro Meeting

1. **Normal End of Sprint Retro**
    a. What went well?
    b. What could have gone better?
    c. Any issues?
2. **Demo**
    a. Playbook Alpha on BC Gov Repo
    b. Solution Scaffolding and News Rendering
3. **Next Sprint**
    a. Possible goals for next sprint
        i. Playbook documentation (contributing)
        ii. Weekly change requests (issues)
        iii. Flux architecture solution scaffolding
    b. Trello board review
4. **Robert's Notes**
    a. How will the team follow the playbook in order to improve the playbook?
    b. Would like to see the 12-factor list filled out as we go to show how we are adopting parts or all of it

c. Explain the solution architecture as we move forward… is it a progressive or responsive web app?
d. Would like the code to move into the BCGOV/CITZ-IMB-Capstone2020 repo
    i. Folder structure
    ii. Readmes/documentation
    iii. The development workflow
    iv. Testing strategy
e. Would like to have a code review on Wednesday June 24

What Went Well?	**What Could Have Gone Better?**
● Please with work output, launching the alpha was awesome! ● When someone completed a task, they moved on to more tasks ● First solution merge is working well ● Great teamwork ● Working within timelines and fluid goals ● People are demonstrating values of the playbook ● Realistic sprint planning and execution	● Time management on Robert's side ● Time management on team side as well

Sprint 2 Retro Notes

Robert and Grant both pleased with productivity.
Grant wants further playbook improvements, expected to happen over the next 8 weeks.
Justin mentioned Boostrap looks like a good framework moving forward, high popularity across Canada.
Grant wants us to get the Flux architecture more fleshed out.
Robert will use GitHub for change requests.
Document how many hours you CAN allocate to a week/sprint, then we can judge how many units we can actually complete.
Keeping track of people's time/availability is very important moving forward.
Need to think of a process we can go through to get from test -> prod.
Code review once a month June 24th, to see what goes into a code review.
Real code review in July.

- make sure we follow best practices.
- Grant wants to be able to follow our logic through comments in our code

- Automated toolsets?
- Security (Sonarcube)

Robert wants to see how we're journalling.
React frameworks to auto-document our code.
Get involved in RocketChat, so people can point us in the right direction.
Start following the playbook
12 Factors
Catalog mistakes.

MURPHY
Document OpenShift/GitHub process.
Document the process of working with the API.

JOSH
Flux architecture needs to be expanded upon.

KELLY
Make corrections to playbook text.

Sprint 1 Retro Meeting

1. **Team Welcome Justin /// Introductions and Roles**
    a. Kelly - Scrum Master, Playbook Main Developer, Team Coach
    b. Josh - UI Developer, Full Stack Developer, Database
    c. Murphy - DevOps, Full Stack Developer, OpenShift
    d. Grant - Technical Steward
    e. Robert - Product Owner
2. **Normal End of Sprint Retro**
    a. What went well?
    b. What could have been better?
    c. Issues?
3. **Show and Tell of What We Have So Far**
    a. Kelly to show the overall Project schedule so everyone knows where we are in our project
    b. Kelly presents the playbook
    c. Kelly to discuss the conceptual news architecture (or what we have of one so far)
        i. Focus is this is evolving as this is the end of our 'experiment' phase
        ii. Starting next week we are in our development phase and because we have not laid down the solution architecture until the UI is designed that gives us the ability to pivot based on what the cross functional development team decides.

d. Josh show the UI prototype(s)
e. Murphy describe the OpenShift plumbing and the pipeline structure

4. **Pre-Sprint 2 Planning discussion**
    a. Robert presents the Product Roadmaps
    b. Kelly to show current Trello and what is on the backlog
    c. Team to give an indication how the Camosun Semester is going

What Went Well?	**What Could Have Gone Better?**
● Highly productive ● High level of communication ● There is some good learning happening ● Requirements are presenting themselves ● Playbook iteration going well	● Feeling unprepared for UI task ● Judging the worth (in time/value) of a task ● Learning how to estimate effort ● Ensuring people are available for reviews and input

Sprint 1 Retro Notes

Journals

We should record errors, confusion in our journals. We learn a lot from failures.

Talk about success and barriers every 2nd sprint?

Keep a journal about a topic (Playbook journal, solution journal, etc…) or individual

OpenShift Pipeline

Grant and Robert will discuss what they want to see from webhooks/pipeline cli and let us know next week

How can we track versioning? If we have a bug, how do we know what version that bug appeared in?

Need a method to go from dev -> test -> prod

Important Notes

Need to research BC Gov accessibility standards

Plan a session for conceptual solution architecture

Think about times for solution and

Sprint 0

What Worked Well?
- Positive and regular communication with team and client
- Handled the camosun side of things very well (finished course requirements well before deadline)
- Very good teamwork
- Delivered project charter on time
- Invested time in learning related to capstone
- Getting buy-in from sponsors
- Getting clearer understanding of deliverables
- Learning how to better organize and manage client and client

What Can We Do Better?
- Meetings are a bit lengthy (come to meeting with agenda, shoot for 15 minutes in a daily standup)
- Could have had more structure

Any Action Items?
-