2022 Internship Summary Document
NSX Feature Flag Management
UI/UX Enhancement for NorthStar NSX Phase 1 Feature Flags
Nancy Xing (Product Design Intern), Lei Lei (Mentor), Alex Nhu (Manager)

Overview

What is Feature flags management?

Feature flags management tool is a SaaS Infrastructure Common Platform within NorthStar NSX that wrap code in flags and turns select functionality on/off at runtime and evaluates based on user properties which enhance workflows in an agile style and CI/CD environment. When building a multi-cloud SaaS service for VMware NSX, some key objectives can be addressed with a "Feature Flag" service.

NorthStar NSX Feature Flag Management project is to envision the user experience enhancement to enabled NorthStar product owners to manage and monitor features and the development teams to facilitate feature experimentation and delivery. The vision behind the project is to improve the in-product feature analytics to detect usage pattern and suggest feature and service upgrade and provide ML based insights. In order to achieve enabling businesses to offer tiered levels of functionality, setting data-driven strategies, and of increasing product releases and engineering velocity with minimal risk.

Key Objectives

1) Selective Feature Rollout to SaaS Customers - enable VMware to dynamically enable/ disable features, with granular control over which subset of features to rollout to desired target of customers or deployments in SaaS

2) SaaS vs On-Premise Differentiation - Enable VMware to control access to SaaS-only features, exposing features only through SaaS while withholding those features from on-premise exposure, while leveraging same codebase and release train

3) Internal Developer Productivity – enable developers to work on the same development branch while maintaining branch stability

4) Error handing and dependency compatibility checks - Enable NorthStar / NSX to identify compatible feature capabilities from the underlying platforms (e.g. NorthStar GM to identify if LM has feature capability, NorthStar UI to know what feature capabilities cannot be exposed due to limitations in LM, so that it can hide those settings from NorthStar UI)

Why does it matter?

Increase product releases and engineering velocity

Enable businesses to offer tiered levels of functionality
Provides more flexibility for the user experience

What are the requirements?

1) **Time duration of the feature**
2) **How feature flag works with product analytics tools like Full Story and Supercollider**
3) **Feature can have multiple Feature Flags**
4) **dev/staging/ production**
   (Tech Preview) roll out / rollback functionality
5) **Selective Rollout Criteria**
   - By organization (only want certain clients to test out some features - specify customer names)
   - By regions (only want to rollout to customers with instance in US - Reason: e.g. UI that is not translated; feature set that have strict regulatory compliance related to those countries and we are not ready to launch to those)
   - By NorthStar Instance (customer can have multiple NorthStar instances which is very specific to NSX capabilities - Reason: they can have instance because of scale problems: 50% workload and 50% in the other; instance for production and instance for test/dev)
   - By compliance (e.g. Federal customers they have regulation that requires big banners “top secret environment”- categorized and cohort by compliance regulation)
   - By mode (VMC SDDC, On-Prem, Maas - environment the client is in, features only suitable for certain mode)
   - By Scale (don’t want to break their environment/ issue immediately turn off the feature)

My Design Process

1. Understanding the Problem
2. User Research and Discovery
3. Competitor Analysis
4. User story & User flow
5. Visual design and iteration

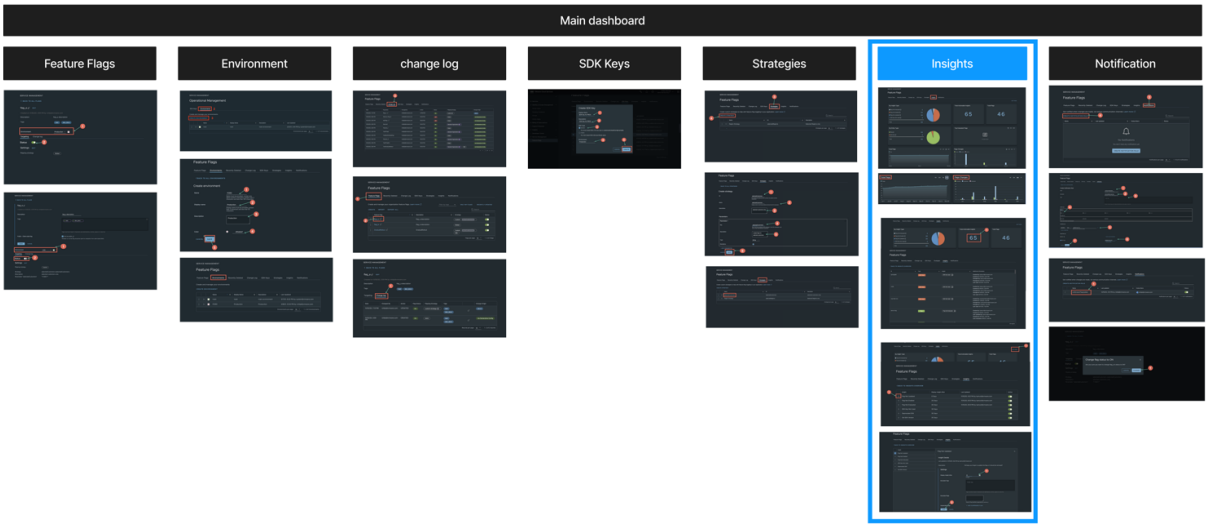
1. Understanding the Problem

For this project, I am taking the CSP Feature flag service, an existing product within VMware and augment the functionality and add new concepts into it.

CSP Feature flag service

CSP Feature flags Service allows VMware Cloud Services developers to take advantage of feature flags technology without the need to set up and maintain a custom feature flag system. It is based on the current organization tenancy model which facilitates integration and management in the context of the service organization. It provides basic rollout strategies, statistic, audit log and notification.

Information Architecture:



CSP insights

In order to gain more visibility on the feature flags configured in the system, CSP provide insights and notifications that will let you manage and get more confidence about your products and features released and their usage.

In the feature flags UI, they have a tab for insights, which will visualize the current insights. The view hold a grid with all latest insight for the organization and filters (by type, insight type, etc.) It is analyzing the on a feature flag specific level which the users can view flag evaluations and usage stats.

For feature flag insights, an opportunity that could be added is to have flag level in-product analytics that tells the customer experience. Product owners are the ones who are going to be consuming any testing results, Insight & Recommendation is for them to have accessible reports that analyze the product and then decide whether to turn the feature on or off and what strategies to set.

There can be possibilities to integrate to have event-based analytics about the product usage and to see the dimension of which was our turn on and off, so we can do A/B tests and things like that. Also, for tech preview features, the state of those would be good to admit into the in-product tracing and metric stream. And can we see how many times a flag was evaluated to

true or like for multi in case of multivariate values, how many times it was evaluated to a particular value.

Design Objectives for Insight
- Detect usage pattern and suggest feature & service upgrade
- Expand on the analytics on the A/B testing of features
- Provide ML based insights
- Showing the dependency and correlation between feature flags

2. User Research and Discovery

Key Persona

NorthStar Product Owners

- Full lifecycle of managing the feature flag
- Create custom user experience
- Discover which different variations of the feature performs better
- Choose who see the feature
- Get feature to users in the real time
- A/B test feature implementation
- Insight- push/poll
- Receive recommendation from the system
- Action- changing the feature flag status
- Pricing model/ outcome of the product
- Continuously experiment and validate on a subset of users
- Measure performance based on the chosen KPIs
- Agile management style and CI/CD environments

NorthStar Developer

- Develop applications and features
- Continuously integrate features into application during development
- See the impact of individual feature quicker
- Make changes without pushing additional code
- Controlled experimentation over the lifecycle of features
- Deploy anytime by separating code deployment from release
- Deliver more feature / deliver feature faster
- Modify system behavior without disruptive changes
- Testing and managing Feature Flags in Dev/Staging

SRE Engineer

- Rolling out and operating-production
- Maintain and Improve reliability, performance and effeciency of the system
- Troubleshoot problems, not exposed to customers

Interview - Understanding the users

To get to know more about the users, I conducted user interviews with product owners, engineers, and SREs. Using the following survey, which helped me to understand the user's current workflow, how feature flag may or may not tie into the current process, as well as understanding the user's desire, experiences and pain points.

Research objectives	Interview Script	Insights & Opportunities
Find out the current experience and use cases with Feature Flag management	Could you briefly describe the experience with CSP Feature Flag management tool and how feature flag management is done right now. Demoing for the flow in which you interact with FF tool	Current workflow for managing feature flag
Find out the background of the user	How frequently do you use this platform? What is the most case scenario to use this? What are your daily actions on this platform?	1) Service Owner for managing the Feature Flags for NSX within VMC on AWS context 2) NS developer 3) SRE team
Find out how users manage and monitor the lifecycle of all flags	How do you manage and monitor total flag overtime and the lifecycle of all your flags across systems? How frequently do you deprecate some of the flags, and does the system have solutions or suggestions for flag retirement?	Managing and monitoring networking features to VMC PM: defining new feature Engineer: creating new feature flags Operation team: turning flags on or off

Find out details about role-based access of the tool	For multiple people who are using the tool, what does the current approval process look like and preventions to avoid breaking someone else's environment? Could you show me your current auditing process and features? Dev/staging/production environment (how it is used currently and how to improve?) How is your SRE team lookup the feature flag insights? Could you share some of SRE's Wishlist and pain points?	Role based access: ticketing mechanism- create a service ticket to request to enable flags
Find out users' experience with Insight & recommendation, and how does it impact on the flag management	How does actionable insight help setting or adjusting the flipping strategy? Could you tell us about a way to roll back to a previous set of feature flags? (Rollback strategy / kill switch) Would integrating with in-Product analytics tool (supercollider / FullStory) help and build reports on top of that help managing the feature flags process? What are some of the most important metrics of the insight and analytics session? What kind of recommendation does the current tool provide?	Currently no telemetry that track and evaluate usage of all the flags, would be helpful to add it on the flag level productivity, user's happiness, satisfaction, rate of errors, support Operate in bundle: having feature flags controlling other flags- easy to scale

Find out existing challenges and actions users take to overcome them	Any other pain point or areas of the tool to be improved on? How did you deal with the challenges? What actions would you have ideally taken, if not already taken?	Text heavy- To have predefined fields: creator just need to click on rather than entering text

Important Meeting Summary

6/23 with Catherine
Project Requirements:

- Granularity of feature flag control
- UI/API access to feature. e.g., only ship the API, hide the UI
- Feature that has multiple feature flags
- Understanding the dependencies and how things are related to each other
- Selective rollout to SaaS customers
- By organization / region / NorthStar instance / compliance / mode / version / scale / support cross service interaction
- Searching and reporting
- How do we make sure that we push features that we want to push out for our customer/ how many flags are live out there (On-prem environment does not have feature flag implemented, only on VMware cloud on AWS)

Next steps:

- Talk to people who used the Feature flag and see if there any downsides and things they struggle with on Feature flags
- We don't have the control of the UI of CSP site (reach out to CSP developer / Srini Seetharaman)
- Explore once we rollout- dashboard of who's using what feature and how they are reacting to the features
- Explore in product feedback & trigger rule- Having opportunities to collect feedback after turning on a feature flag
- Looper - sales team got feedback from customers and sending a request form post subscription (reconsider if it is in scope of the project)

7/12 with Catherine
Before poster session (Aug 3)
Polish on current prototype + finishing the end-to-end flow for the flowing areas:
Insight

- Edition on the active user chart: Y-axis-maximum possible active users
- Customer based view: query on customer- Integrate in-product feedback survey/

Full Story

- Show Satisfaction rates / Conversion rates / Task completion rate
- Visualize the performance impact for particular feature flags
- Insight breakdown per customer (SRE)

Recommendation

- Difficulty of retiring feature flags - show which ones to deprecate (maybe include in this session)
- Reduce amount of flag (simplicity operation)
- Enhancements on dependency Chart- what related feature to turn on / timeline for each feature
- Explore on rollback - how/ when to bring the feature back

Conduct Interview - VMC:

- Capture use case & real data
- How they manage and monitor total flag overtime
- Alert (What action they see after receiving the alert)
- Inquire on Role-base access to the management tool
- Current approval process
- Dev- staging- production environment (how it's used currently and how to improve for NorthStar)

After poster session
Research + Exploration:
Audit log

- Add more level of details on who changed what, where, when, why, which feature flag broke it
- What specifically updated / what new rollout strategy - details
- Avoid breaking someone else's environment
- Approval process- integrate with audit log (every time a new flag implemented submit that to approval)

7/21 with Sandeep
User Background

Sandeep Sharma- Service Owner for managing the Feature Flags for NSX within VMC on AWS context

Current Use Cases

- Managing and monitoring networking features to VMC
- PM: defining new feature
- Engineer: creating new feature flags
- Operation team: turning flags on or off
- Role based access: ticketing mechanism - create a service ticket to request to enable flags (common SaaS Service)

Opportunities

- Currently mostly plain text and need to enter everything manually
- add predefined fields so creator just need to click on rather than entering text
- Operate in bundle: having feature flags controlling other flags- easy to scale -
- create dependency in feature flags
- Telemetry: track and evaluate usage of all the flags

8/10 with Tea
Areas to work on for the project:

- Deep dive into some use cases

To take it further: e.g. How does infra_classification tie into feature flag?

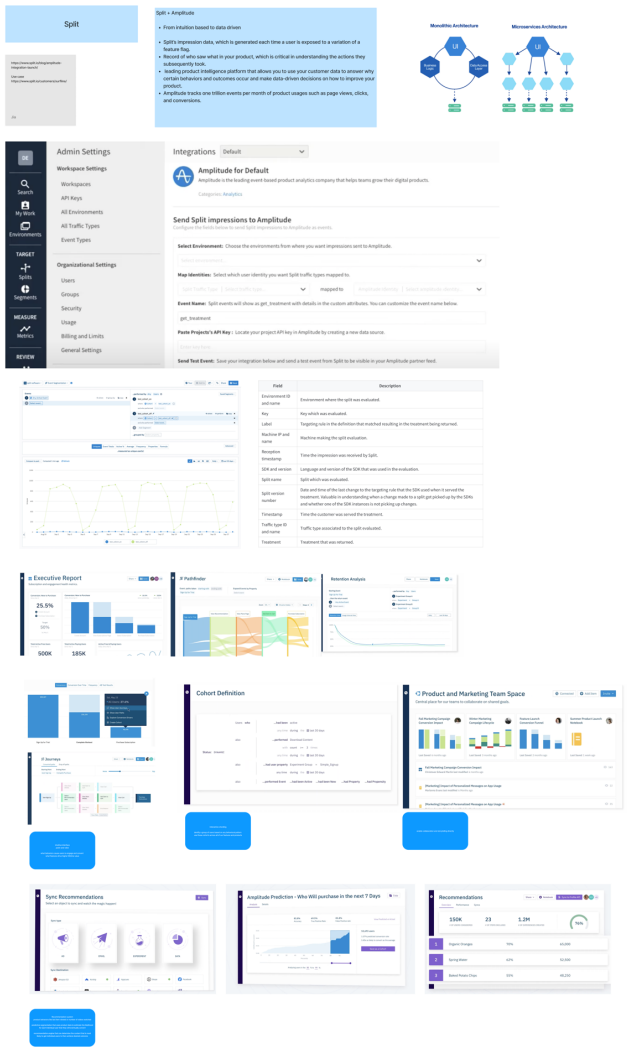
- What do users want to learn from the product analytics in terms of the feature flag?
- The process is automated: How accurate (accuracy rate) they are? Is our algorithm improving? track how many manual actions users must take
e.g., rollout different versions of the same thing (2 or 3 variations), and compare which variation is taking less user intervention
- For the enterprise product - the less interaction the better. For user engagement is questionable because we don't want customers to spend a lot of time on the UI and be as quick as possible, which is completely different from the consumer side of things
- Is there is not variant testing, why would be the purpose of combining feature flag and product analytics?
- For recommendation, evidence and transparency are very important: why made the recommendation and is this recommendation that I can rely on
e.g., we saw anomaly somewhere, that's why we believe this is malicious behavior
-Work more on the detailed end-to-end use case

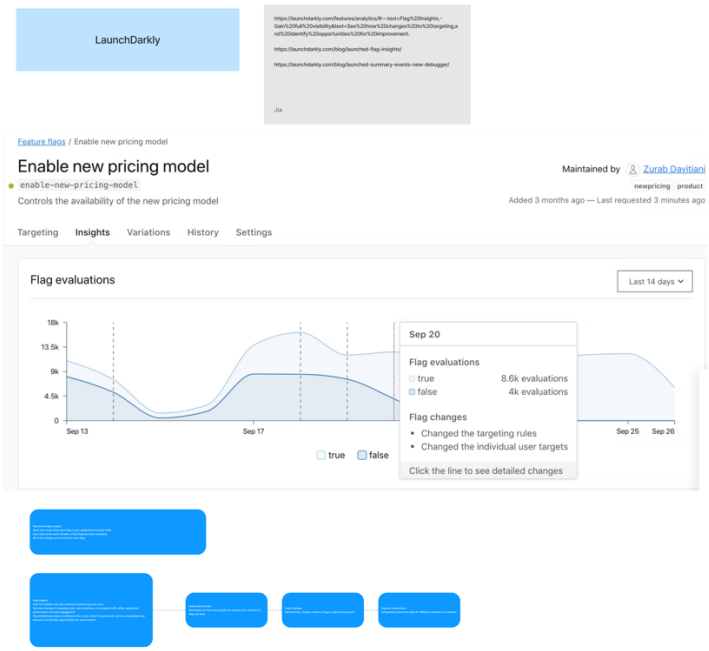
8/22 with SRE team
SRE ff Panel:

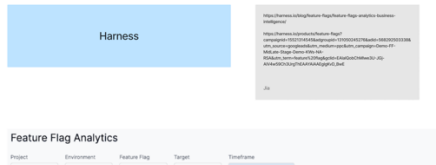
- Number of service customer subscribed / How many features they have enabled throughout the time
- Feature- related to incident
- Integrate into satisfaction rate
- Feature dependency chart
- Describing feature used by what customer/ what time

Alarm - instance- incident - security/ latest incident- when
Priority level - how long has been opened - trouble shooting behind the customer
Business level - ability for them to search fast - hundreds of incidents- filter out the
Specific individual Customer based view on the product usages
SRE goals:
Quick find what they find and identify
Minimal step for is to trouble shot
Grouping - SRE - keep everyone up to date - tackle the incidents they have

3. Competitor Analysis







For competitive analysis I have evaluated on seven different platforms summarized in the below document. Specially on the insight and recommendation session:

https://www.figma.com/file/R00YpWh2Fc7znTyBxDQD8B/Feature-Flag-Research?node-id=0%3A1

Components from the competitor's insight session which could be integrated:

Audit changes:
Show how many times each flag in your application has been activated
How many times each variation of that flag has been evaluated
Review every change made to a flag in a given time period

User behaviors:
See trends on how many people are seeing each version of a flag over time
See how changes to targeting rules, new variations, or increased traffic affect application performance and user engagement

Cohort:
identify a group of users based on any behavioral pattern
Predictive segmentation that uses product data to estimate the likelihood for each individual user that they will eventually convert
recommendation engine that can determine the content that is most likely to get individual users to then achieve desired outcome

Test evaluation:
Comparing conversion rates for different variations of a feature
Flag evaluation result/reason
Flag usages in your code (components)
Flag evaluation over time, grouped by component/eval-reason/flags, etc.

Recommendation:
Generate a list of items to recommend based on users perform prior to the conversion event increase conversion with automated machine learning
Build new recommendation based on the desired outcomes and user segments
Select and apply the appropriate insight/action plan pairs, and measure their impact against KPIs.

Thought process: integrate some parts into the current insight session
Higher level breakdown

Detailed breakdown
Overview
High-level statistics dashboard (visualization enhancement: content hierarchy, color code)
Total insights/ Total Flags
Summary by "insight type"/ Summary by "insight entity"
Top evaluated flags / SDK type
Total flags overtime / Flag Changes
+Feature Dependency

Insight
Audit changes: Review every change made to a flag in a given time period
Active User: See trends on how many people are seeing each version of a flag over time
User behaviors/feedback: See how changes to targeting rules, new variations, or increased traffic affect application performance and user engagement
Cohort: identify a group of users based on any behavioral pattern
Recommendation: 1. ML based recommendation 2.Build new recommendation based on the the desired outcomes and user segments
Recommendation evaluation

4. User Stories & User flow

User Stories
For product owner:

User Story 1
- create a flag with three variants (3 UI style) going to parallel market.
- Rollout the three versions and compare the results: task completion rate, accuracy and cancellation. Also see how many times user come and modify the policies, and where they made the changes.
- Recommendation on which variation is the strongest, and users can click to see the real numbers behind them. Rollback the other flags

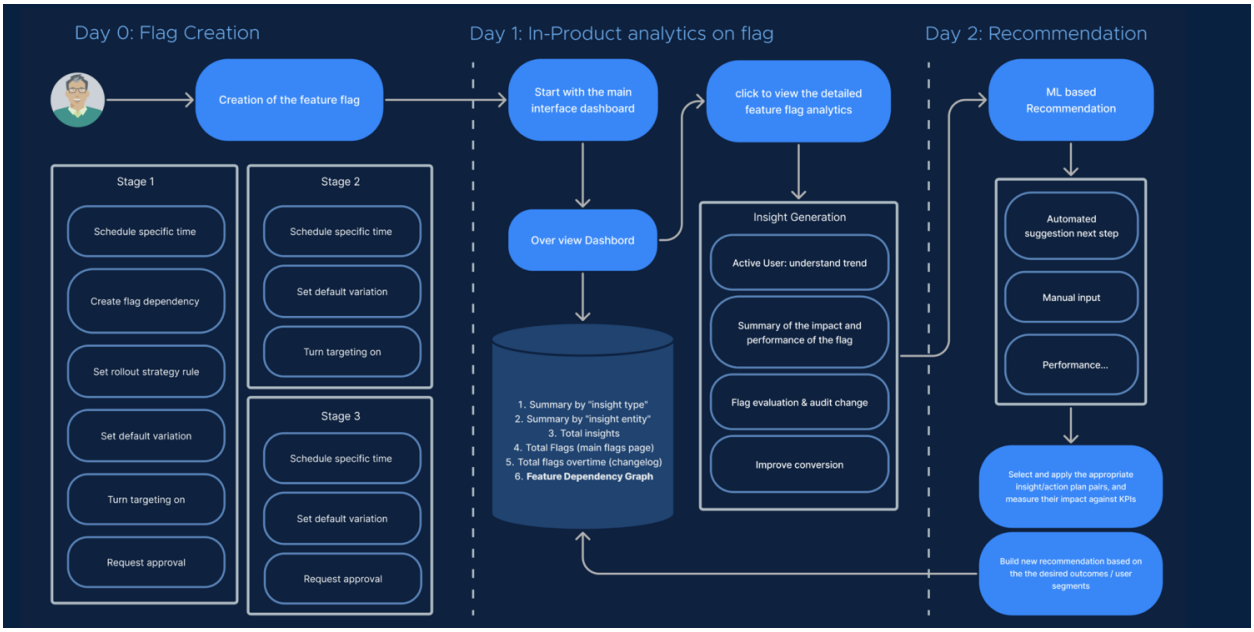
User story 2
- We're doing some usability A/B testing. For first 7 days, user will access the feature through UI-style-A, with feature flag indicating Jan 7 to be the end date of UI-style-A.
- For next 7 days, user will access the same feature through UI-style-B. At the end of 14 days, the feature will be disabled due to the end of our usability test session.

For SRE:
- leverage feature flags as a way of supporting the core goal: Maximizing reliability.

- you discover that the latest release of an application you support has a severe latency problem due to an issue with a new networking feature in the app.
- If that feature is controlled via a feature flag, you can remediate the issue very quickly by simply switching it off. You don't have to wait for developers to fix the underlying code issue before you can solve the problem for your users.

User flow

The User flow is a series of actions a user takes to reach the goal.



Day 0

Multi-stage Flag creation

Stage 1

- Schedule specific date and time
- Add pre-requisite flag/ create flag dependency
- Set rollout strategy rule (organization, Region, NorthStar Instance, Compliance Regulation, Mode, Version, scale, percentage etc.)/ create targeting rule
- Set default variation (multivariant flag: UI-style-A / UI-style-B)
- Turn targeting on
- Request approval

Stage 2

- Schedule specific date and time
- Set default variation (multivariant flag: UI-style-A / UI-style-B)
- Turn targeting on

Stage 3

- Schedule specific date and time
- Turn targeting off

Day 1
View Dashboard
High-level statistics of all flags

Total insights (Insights) Summary by "insight type" / Summary by "insight entity"
Total Flags (leading to main flags page)
Total flags overtime (changelog)
Flag Dependency Chart

Insight & Analytics on one specific flag
User behaviors:
See trends on how many people are seeing each version of a flag over time
See how changes to targeting rules, new variations, or increased traffic affect application performance and user engagement
Satisfaction rates / Conversion rates / Task completion rates

Test evaluation:
Comparing conversion rates for different variations of a feature
Flag evaluation result/reason
Flag evaluation over time, grouped by component/eval-reason/flags, etc.
View audit log

Cohort:
identify a group of users based on any behavioral pattern
Predictive segmentation that uses product data to estimate the likelihood for each individual user that they will eventually convert
recommendation engine that can determine the content that is most likely to get individual users to then achieve desired outcome

Day 2
Recommendation & Actionable Insights

Generate a list of items to recommend based on users perform prior to the conversion event increase conversion with automated machine learning

Build new recommendation based on the desired outcomes and user segments

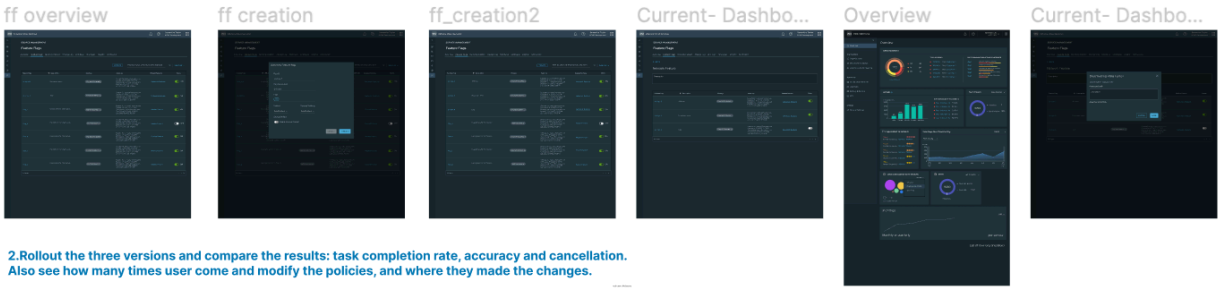
Select and apply the appropriate insight/action plan pairs and measure their impact against KPIs.

View Audit log

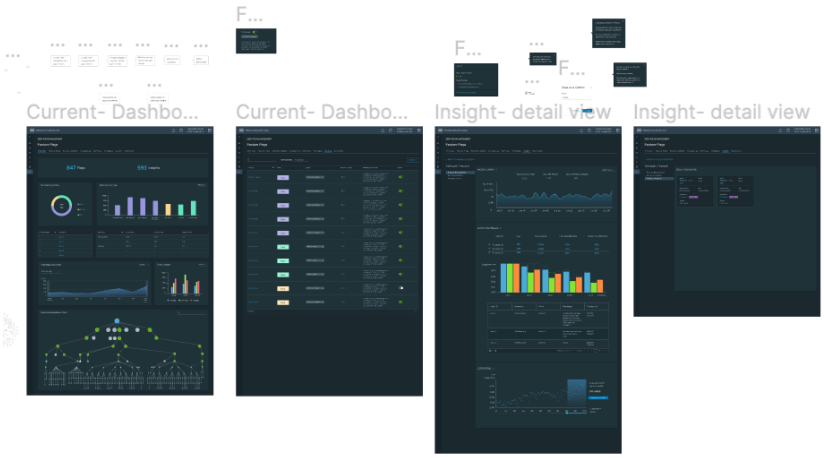
- Details on who changed what, where, when, why, which feature flag broke it
- What specifically updated / what new rollout strategy - details
- Avoid breaking someone else's environment
- Approval process- integrate with audit log (every time a new flag implemented submit that to approval)

5. Visual design & iteration

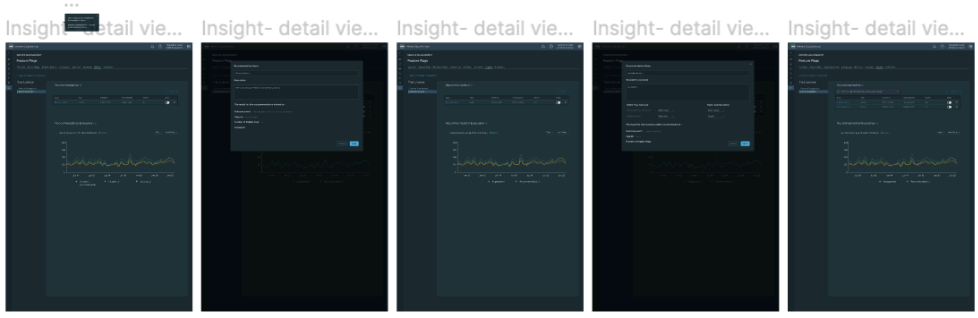
1. create a flag with three variants (3 ui style) going to parallel market.



2.Rollout the three versions and compare the results: task completion rate, accuracy and cancellation. Also see how many times user come and modify the policies, and where they made the changes.



3. Recommendation on which variation is the strongest, and users can click to see the real numbers behind them. Rollback the other flags



Multiple iteration of prototypes:
https://www.figma.com/file/c06TmJ7pI6Q8QfEDWHLKYS/prototype?node-id=963%3A48860

Project poster presentation:
https://insidertv.vmware.com/media/t/1_h48v8qj8/265003862

Conclusion

What did I learn from this internship?

When I started my internship on June 6th, 2022, I realized that I have mountains to climb. At each phase of the project, I learned new concepts about the product, explored different areas, investigated on the user needs for different stakeholders, and enjoyed every step taken to complete this project. This is my first time working on enterprise product project from scratch starting from the research to prototyping, I was navigating in the unknown, but always feel engaged because of the time that I invested to learn and explore new knowledge. I completely enjoy the journey of designing this project, and it is a priceless experience to learn from the NSX UX design team.

Key takeaways:

- Understanding the users is the key to start making a great product.
- The less interaction the better. For enterprise product we don't want customers to spend a lot of time on the UI, which is different from the consumer products
- Constant feedback, critiques and Iterations are important parts of the process.
- There is always room for improvement.
- For UI design, follow design pattern and have more clear visuals
- Constructing detailed user cases could help in telling the story
- More confident communication skill with users and other designer or stakeholders
- More systematic way when giving a presentation

(Background story- Challenges- solution- design)

- Ability to work in multidisciplinary team: worked not only with designers, but also engineers, SRE, PMs to understand their perspective. Also, the Hackathon event is helpful to hone skills of communication and collaboration with others.
- What I want to improve further: presentation skills and visual UI design