>>Project Name<<	Gap Analysis
Context	
Business Context	**Creating an Error Handling Solution for The Exchange internal Capital One product.**
Problem or Opportunity Statement	**By integrating Error Handling, The Exchange will cut down on support time and decrease onboarding time and therefore improve experience for our customers.**
Goal	**Integrate Error Handling within next 2 PIs and help all Capital One teams onboard to the Exchange by 2022.**
Success Criteria	
Error Handling should be implemented at every submission page during onboarding process	
Error Handling should project errors from AWS logs to the UI for the Exchange	
Create a team of 5 engineers (hire missing team members)	
Document all error messages and explanations used for Error Handling for Support	
Choose correct error messages during the team meeting by involving downstream systems: Snowflake and Kafka	
Create training meetings for Support team	
Test 90%+ edge cases for the functionality	
Deliverables or Requirements	**Description**
Design and document steps to pull error messages from AWS logs	Pull an engineer to design the solution on the back end to pull log messages. Design: 1 sprint, Implementation: 1-2 sprints
Design and document user friendly messages to project to UI	Pull an engineer to design the solution on the front end to rewrite messages for users. Design: 1 sprint
Design and document steps populate error messages to front end	Pull an engineer to design the solution to project messages from backend to the frontend. Design: 1 sprint, Implementation: 1-2 sprints
Create Step Functions to represent a flow of errors	Create AWS Step Function solution with 3 lambdas to have reusable solution for future functionality. Implementation: 2 sprints, 2 engineers
Document message meaning for Support	Document all error messages and their descriptions in a user friendly way. Documentation: 1 sprint
Zoom meeting with Support team	Schedule a zoom meeting and a follow up meeting with Support team to discuss new functionality
Create automated tests for error handling	Design and write automated tests for new functionality to cover at least 90% of intent. Implementation: 2 sprints
Run automated tests	Run automated tests and make sure they all pass before release. Implementation: 1 sprint
Call with Snowflake and Kafka teams	Schedule zoom meeting with Snowflake and Kafka downstream teams to discuss error messages and meaning. 1 sprint
Release to Prod	Release new functionality to Production environment
Create a feedback form for customers	Create a feedback form and release it to users after a month of new functionality released.

Error Handling for the Exchange	**Gap Analysis**		
Current State	**Future State**	**Gap**	**Requirements**
People			
Customers are confused about what is going wrong with their onboarding form	Customers have a clear understanding on what the error means and what steps they can take to fix it	The understanding of what the error means and what the actual error is instead of a general broad message.	Create a translation for customers of log errors and project them to the UI
Customers have to reach out on average 3 times to support team to understand what's wrong with the form	Customers reach out on average less than 1 time to support per dataset onboarded with questions	Customers reach out more often than the stakeholders would like to the Support team (3 times more)	Create office hours to teach customers new functionality Clearly document all errors
No clear documentation on what errors mean to customers	Customers have clear guidelines documented on Capital One shared space	No documentation presented on errors in the Exchange to customers	Create and share documentation regarding all errors in the exchange
Process			
The process of one team onboarding take longer than 1+ sprint with all error they face	The team take less than a sprint to onboard a dataset to the exchange	Customers take on average twice as long to onboard than their estimate their time to due to errors	Create a readable error handling solution
Each team requires personal assistance from Support to get onboarded	Only most unique cases or bugs in the system require support	Support should be used for unusual errors and bug fixes, not general support	Utilize office hours and documentation with readable error messages to fix repeating questions
Technology			
There is no reusable error handling solution for future functionality	There is a step function solution with reusable aws lambdas for error handling	Current solution is not scalable and not using new fast growing aws technology	Create Step Functions with 3 lambdas for the design Document steps on how to reuse lambdas for future functionality
Team is not pulling errors from AWS logs	AWS logs are widely used within the Exchange design	Engineers do not utilize their aws logs to solve the problem	Research and design solution to pull logs from aws to backend of the Exchange