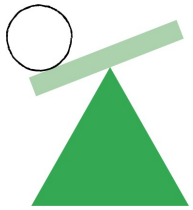
# Errors + Grace Failure
# Chapter worksheet



## Instructions

Block out time to get as many cross-functional leads as possible together in a room to workthrough these exercises & checklists.

## Exercises

### 1. Error audit [~1 hour]

Collect canonical error examples to define existing and potential errors and solutions.

### 2. Quality assurance [~30 minutes]

Prioritize how you'll test and monitor errors and reporting so you can hear from your users early and often.

# 1. Error audit

As a team, brainstorm what kinds of errors users could encounter. If your team has a working prototype of your feature, try to add current examples.

Use the template below to start collecting error examples so your team has a shared understanding about the different error types and solutions your model could produce.

| **Error** | **Users** |
| --- | --- |
| Inaccurate weather prediction due to data inconsistency | |
| **Error type** | **User stakes** |
| ☐ **Background** - Situations in which the system isn't working correctly, but neither the user nor the system register an error. | ☐ high |

| **Error** | **Users** |
| --- | --- |
| Chatbot response time exceeding 2 seconds | |
| | **User stakes** |
| | ☐ high |

| Error type | |
|---|---|
| ☐ **System Limitation -** Your system can't provide the right answer, or any answer at all, due to inherent limitations to the system. | |

| Error | Users |
|---|---|
| System failure during peak load times | |
| **Error type** | **User stakes** |
| ☐ **System Limitation -** Your system can't provide the right answer, or any answer at all, due to inherent limitations to the system.. | ☐ high |

| Error | Users |
|---|---|
| Data pipeline interruption due to API failures | |
| **Error type** | **User stakes** |
| ☐ **Background** - Situations in which the system isn't working correctly, but neither the user nor the system register an error. | ☐ Low |

## Error sources

Take each error identified above through these questions to determine the source of the error:

## Input error signals

- ☐ Did the user anticipate the auto-correction of their input into an AI system? Yes, for chatbot interactons
- ☐ Was the user's habituation interrupted? If the system provides unexpected recommendations
- ☐ Did the model improperly weigh a user action or other signal? if there's a context error in interpreting user preferences for weather-based recommendations.

# Relevance error signals

- ☐ Is the model lacking available data or requirements for prediction accuracy?

Possible, especially if there are gaps in real-time data from APIs.

- ☐ Is the model receiving unstable or noisy data?

Yes, this is a potential issue with real-time weather data.

- ☐ Is the system output presented to users in a way that isn't relevant to the user's needs?

Possible, if personalized recommendations are not accurately tailored.

## System hierarchy error

- ☐ Is your user connecting your product to another system, and it isn't clear which system is in charge?

  Not applicable in this case.
- ☐ Are there multiple systems monitoring a single (or similar) output and an event causes simultaneous alerts?

  Possible, given the multiple data sources and monitoring tools in use.

## Failure state

- ☐ Is your feature unusable as the result of multiple errors?

Possible, if there are cascading failures in the data pipeline or model prediction accuracy.

# Error resolution

Once you have identified the source or sources of the error, complete the sections below for each of the errors in the template with your team's plan for improving / reducing the identifiederror: Create as many copies as you need to cover all your identified errors.

| **Error rationale** | **Solution type** |
| --- | --- |
| Why the user thinks this is an error:<br><br>Inaccurate weather prediction due to data inconsistency<br><br>The weather prediction doesn't match the actual conditions they observe | ☐ Feedback |

## Error resolution

User path:

Examples: User sees errors, gives feedback, completes task; User sees error, takes overcontrol, completes task

Opportunity for model improvement:

Example: User's feedback logged for model tuning

# 2. Quality assurance

Getting your feature into users' hands is essential for identifying errors that your team, as expert users, may never encounter. Meet as a team to prioritize how you want to monitor errors reported by users so that your model is being tested and criticized by your users early and often.

As you have this discussion, consider all potential sources of error reporting:

- Reports sent to customer service
- Comments and reports sent through social media channels
- In-product metrics
- In-product surveys
- User research (out-of-product surveys, deep dive interviews, diary studies, etc.)

## QA template

| **Goal**<br><br>**Monitor Prediction Accuracy** | **Review frequency**<br><br>☐ Daily |
|---|---|
| **Method**<br><br>Use MLflow to track model performance metrics and set up alerts for accuracy drops below 90% | |
| Start date:<br>Review / End date: | |

| **Goal**<br><br>**Review Frequency** | **Review frequency**<br><br>☐ Daily |
|---|---|

| | |
|---|---|
| **Method**<br><br>Use Airflow's built-in monitoring to track DAG task failures and completion times | |
| Start date:<br><br>Review / End date: | |

| | |
|---|---|
| **Goal**<br><br>**Monitor Data Pipeline Integrity** | **Review frequency**<br><br>☐ Daily |
| **Method**<br><br>Use Airflow's built-in monitoring to track DAG task failures and completion times | |
| Start date:<br><br>Review / End date: | |

| | |
|---|---|
| **Goal**<br><br>**Monitor System Performance** | **Review frequency**<br><br>☐ Daily |
| **Method**<br><br>Use Prometheus and Grafana to track CPU, memory, and network usage of Kubernetes clusters | |
| Start date:<br><br>Review / End date: | |

| Goal | Review frequency |
| --- | --- |
| Monitor user feedback and satisfaction | ☐ Weekly |
| **Method**<br>Analyze in-product surveys and user research data | |
| Start date:<br>Review / End date: | |