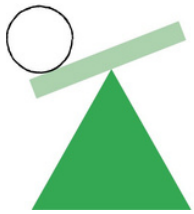
Errors + Grace Failure
Chapter worksheet



Instructions

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

Exercises

1. Error audit [~1 hour]

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

2. Quality assurance [~30 minutes]

Prioritize how you'll test and monitor errors andreporting 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.

Error: API /Data feed failures and Outdated Data

INFO:fetch_bluebikes_data: Fetching station data...
ERROR:fetch_bluebikes_data: HTTP 404 - Endpoint not found for date range

Error type

- [x] **System limitation**- Your system can't provide the right answer, or any answer at all, due to inherent limitationsto the system.
- [x] **Context**- The system is “working as intended,” but the user perceives an error because the actions of the systemaren't well-explained, break the user's mental model, orwere based on poor assumptions.
- [ ] **Background**- Situations in which the system isn'tworking correctly, but neither the user nor the system register an error.

Users

User stakes

- [x] low
- [ ] high

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 theirinput into an AI system?
- [x] Was the user's habituation interrupted?
- [ ] Did the model improperly weigh a user action or othersignal?If yes, likely a context error.

Relevance error signals

- [x] Is the model lacking available data or requirementsfor prediction accuracy?
- [ ] Is the model receiving unstable or noisy data?
- [x] Is the system output presented to users in a way thatisn't relevant to the user's needs?

System hierarchy error

- [ ] Is your user connecting your product to another system,and it isn't clear which system is in charge?
- [x] Are there multiple systems monitoring a single (orsimilar) output and an event causes simultaneous alerts? We have multiple ways of monitoring this sort of issue

Failure state

- [ ] Is your feature unusable as the result of multipleerrors?

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 identified error: Create as many copies as you need to coverall your identified errors.

Error rationale	**Solution type**
The user cannot obtain the expected results. This makes sense because the user wants to see predictions regarding the number of bikes, but instead is met with technical gibberish about an API. Conversely, the user noticed that a fully occupied bike rack was prompted to be restocked, which makes no sense. This occurred because the data used by the system was old or corrupted.	● Feedback ☐ User control ☐ Other:

Error resolution User path: User noticed broken output

In an ideal situation, the APIs will always function correctly and the data feed will flow continuously. Our pipeline will then generate a model capable of providing the user with the expected results. However, if the APIs fail or the data feed is interrupted, the pipeline should be robust enough to ensure that the user is not negatively impacted — or, at the very least, is informed about what went wrong and that the issue is being addressed.

Furthermore, it is possible that the user may encounter inaccurate results caused by delayed or outdated data being processed. In such cases, the user should have a clear channel to report inaccuracies or provide feedback.

In summary, proper health checks should be implemented to monitor data and API reliability, and user feedback channels should be available to ensure transparency and continuous improvement.

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 examplesso your team has a shared understanding about the different error types andsolutions your model could produce.

Error: API or Data feed failures Innacurate Predictions and Model Drift	**Users** **User stakes** ☐ low ● 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. ● **Context**- The system is “working as intended,” but the user perceives an error because the actions of the systemaren't well-explained, break the user's mental model, orwere based on poor assumptions. ● **Background**- Situations in which the system isn'tworking correctly, but neither the user nor the system registeran error.	

Error sources

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

Input error signals

- [ ] Did the user anticipate the auto-correction of their input into an AI system?
- [ ] Was the user's habituation interrupted? Did the model improperly weigh a
- [ ] user action or othersignal?If yes, likely a context error.

Relevance error signals

- [ ] Is the model lacking available data or requirements for prediction accuracy?
- [ ] Is the model receiving unstable or noisy data?
- [ ] Is the system output presented to users in a way that isn't relevant to the user's needs?

System hierarchy error

- [ ] Is your user connecting your product to another system,and it isn't clear which system is in charge?
- [ ] Are there multiple systems monitoring a single (orsimilar) output and an event causes simultaneous alerts?

Failure state

- [ ] Is your feature unusable as the result of multipleerrors?

Error resolution

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

Error rationale	**Solution type**
Unbeknownst to the user, the prediction is inaccurate due to internal issues—either overestimation or underestimation of bike demand. This may stem from underlying logic errors or from model drift, where changing patterns cause shifts in data correlations. Without regular retraining and continuous monitoring of forecast accuracy, the model's predictions will lose relevance, ultimately leading users to lose trust and stop using the product.	☐ Feedback ☐ User control ● Other:

Error resolution

Internal checks should be implemented to continuously monitor the accuracy of predictions, user satisfaction, and shifts in data patterns or correlations. These checks would help identify when the model begins to deviate from expected performance or when external factors—such as seasonal demand changes, user behavior shifts, or data quality issues—start to affect outcomes.

Regular evaluation metrics, such as prediction error rates, user feedback trends, and data drift indicators, should be logged and reviewed. If significant deviations are detected, automated alerts can trigger model retraining or manual review to ensure the system remains reliable and responsive to real-world changes.

2. Quality assurance

Getting your feature into users' hands is essentialfor 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 testedand criticized by your users early and often.

As you have this discussion, consider all potentialsources 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 Ensure the reliability, accuracy, and user relevance of Bluebikes Demand Prediction by contnuously validating model outputs, data integrity and user satisfaction.	**Review frequency** ☐ Daily ● Weekly ☐ Monthly ☐ Other:
Method Use automated validaton checks e.g. predicted vs actual bike demand, trigger alerts if deviation thresholds are exceeded, run airflow health checks, categorize and analyze user feedback, track key performance indicators.	
Start date: 10/28/2025 Review / End date: 10/28/2026	