Errors + Grace Failure

Chapter worksheet

Instructions

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

Exercises

1. Error audit [~1 hour]

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

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	User(s)	User Stakes	Error Type	Description / Example
1. Missing or corrupted dataset files in DVC	Developers / Evaluators	High	System limitation	DVC pull fails due to missing remote version or corrupted .dvc files.
2. Schema mismatch during data merge	Data engineers	High	Background	"Prompt" or "Category" column names differ between datasets (e.g., SALAD vs. internal datasets). Causes join failure or null rows.
3. Wrong category mapping or exclusion rules applied	Evaluator / Judge	Medium	Context	Child Sexual Abuse category excluded mistakenly due to incorrect filter logic.
4. DVC remote connection failure (GCP bucket)	Developer	High	System limitation	DVC can't connect to GCP due to expired credentials or misconfigured remote.
5. Model version mismatch	Judge / Evaluator	High	Context	Evaluator API uses old model version (e.g., stale cached container image). Results inconsistent with latest run.
6. Data leakage or incorrect split	ML engineer	High	Background	Training/test data overlap due to random seed inconsistency.

Error	User(s)	User Stakes	Error Type	Description / Example
7. Incorrect prompt text length calculation	Evaluator UI or API logs	Low	Background	Word count function fails for unicode or special tokens.
8. Pipeline execution halted due to dependency failure	All	High	System limitation	Docker container (evaluator-api or judge) fails to start due to missing environment variable or image version mismatch.

Error sources

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

Error	Likely Source	Explanation
Missing/corrupted DVC files	Input error + relevance	Remote data missing; improper DVC push; human error.
Schema mismatch	Context	Poor documentation or version drift between datasets.
Wrong category mapping	Context	Human misunderstanding of inclusion/exclusion logic.
GCP DVC failure	System hierarchy	Ambiguous ownership between DVC + GCP credentials.
Model version mismatch	System hierarchy	CI/CD didn't propagate the latest Docker image.
Data leakage	Relevance	Split script not deterministic or seed not fixed.
Prompt length miscount	Input	Edge cases (e.g., newline tokens).
Pipeline dependency failure	System limitation	Docker-compose misconfiguration or missing dependency.

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 other signal? 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 (or similar) output and an event causes simultaneous alerts? Signal crashes increase the user's mental load because they have to parse multiple signals to figure out what happened and what to do next.

Failure state

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

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 cover all your identified errors.

Error	Why User Thinks It's an Error	Solution Type	User Path	Opportunity for Model / System Improvement
DVC pull fails	Pipeline not reproducible	Feedback + Control	User retries with error logs visible	Add DVC remote check pre-flight; integrate dvc doctor in CI.
Schema mismatch	Data processing broken	Feedback	Pipeline fails with schema diff shown	Schema validation step before merge.
Category mapping issue	Output dataset missing category	User control	User edits config and reruns	Maintain category config JSON under DVC.
DVC-GCP auth failure	Can't pull data	Feedback	Display clear auth message	Add credential refresh workflow + fallback to read-only public bucket.
Model version mismatch	Inconsistent results	Feedback + Control	User triggers version check	Implement MODEL_VERSION tag in metadata tracked in DVC.
Data leakage	Unreliable evaluation	Other	Auto-validation script runs before model training	Log and block commit if leakage detected.
Prompt word count	Metrics off by small margin	Feedback	Warning in logs	Add test cases for multilingual text.
Dependency failure	Containers not starting	Feedback	Error in compose logs	Health check in docker-compose, automatic rollback.

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	**Review Frequency**	**Method**	**Start Date**	**Review / End Date**
Ensure dataset reproducibility and model traceability	**Weekly**	Run DVC pull + dvc status, verify model hash matches metadata	10/28/2025	Continuous
Detect schema or merge issues early	**Daily (during dev)**	Automated schema validation in CI	10/28/2025	Continuous
Monitor GCP remote reliability	**Weekly**	Check credentials, storage latency, and logs	10/28/2025	Continuous
Model version audit	**Monthly**	Verify evaluator + judge Docker images	10/28/2025	Continuous
User feedback loop (Evaluator UI / Judge)	**Weekly**	Error reports logged via API	10/28/2025	Continuous