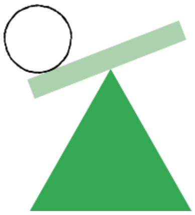
Errors + Grace Failure

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.

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 Type	Source	Severity
The RAG system fails to surface a relevant section	*Data coverage limitation* — the metric may not appear in filings (companies report differently), or the preprocessing pipeline failed to extract/standardize it. In RAG systems, this can also occur if the retrieval model does not match equivalent phrasing (e.g., “Earnings before interest, taxes, depreciation, and amortization” ≠ “EBITDA”).	**High** — Missing key metrics can lead analysts to conclude the data is absent or incomplete, potentially affecting trust in the system or forcing manual re-checks in SEC filings.
Truncated Responses	The AI only summarizes part of a section (e.g., the first few paragraphs of “Liquidity and Capital Resources”) due to token length constraints.	**Low** — Partial information requires re-querying, slowing workflow but not invalidating results.
Outdated Data Retrieval	The model pulls an older version of a company’s 10-K when newer filings exist.	**High** — Analyst decisions may rely on obsolete information.

Error sources

Errors + graceful failure

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?

□ 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 rationale

Why the user thinks this is an error:

- **RAG system fails to surface a relevant section:**

  The system provides a summary or answer that feels **too generic** or **unrelated** to the user's query. The retrieved section does not align with the **intent of the financial question** (e.g., when asking about liquidity, the response pulls "revenue trends" or "operating segments" instead).

- **RAG system provides truncated responses:**

  The answer stops mid-sentence or lacks closure ("The company's liquidity improved due to… [cut off]"). It doesn't address all parts of the question (e.g., only explains 2022 results but skips 2023). Users expect **complete, contextually coherent responses** but receive **fragmented or partial summaries**, reducing trust. Inconsistent behavior — some queries yield full responses, others are cut short, even for similar topics.

- **Outdated data retrieval**

The user expects **current and accurate** financial insights, especially for comparisons or recent performance.The system sometimes quotes outdated metrics or old 10-K sections, leading users to believe the model is **inaccurate or unreliable**.Users may lose confidence if the model provides **stale insights**, especially when newer filings exist.

Solution type

☐ Feedback
☐ User control
☐ Other:

Error resolution

User path:

Scenario:
The user runs a financial query (e.g., “Compare Coca-Cola’s latest operating income with PepsiCo’s”) and notices the model surfaces FY2020 or FY2021 data instead of the most recent filing.

- **User Reaction / Flow:**

1. The user **recognizes the outdated information** (e.g., date mismatch in response).
2. The user **flags the issue** using in-product feedback (thumbs-down, error report form, or comment box).
3. The user may **manually check** the SEC EDGAR database to confirm the latest filing.
4. The user **rephrases the query** (“based on FY2023 report”) to attempt correction.
5. The user ultimately **completes the task manually** or partially relies on model output.

- **Outcome:**

The user loses partial trust in model recency but continues using the product if feedback acknowledgment and visible improvements occur quickly.

Opportunity for model improvement:

- **Feedback Logging and Routing:**
- The system captures user feedback tagged with the query, timestamp, and filing year mentioned.
- This data is logged into an **“error feedback dataset”** for post-analysis and retraining.
- **Retraining / Fine-tuning Signals:**
- Model improvement team reviews feedback clusters (e.g., “outdated data”) weekly.
- Signals are used to **retrain retrieval filters** or adjust **metadata weighting** (to prioritize newer filings).
- **User-Informed System Adaptation:**
- When users re-query with date-specific terms, the system learns to **boost recent-year embeddings** automatically.
- Future retrievals leverage **reinforcement-like feedback**, improving contextual awareness of “latest” vs “historical” intent.

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 Ensure the AI-driven 10-K analysis system (both qualitative RAG responses and quantitative metric extraction) produces accurate, interpretable, and user-trusted results. Continuously monitor user-reported errors and model drifts to improve reliability and user confidence over time.	**Review frequency** ✓ Monthly
Method • User Error Monitoring: Collect reports from users via in-product feedback (flag button or "Report issue" feature). • In-Product Metrics: Log failed queries, empty retrievals, or low similarity scores from the RAG pipeline. Monitor quantitative accuracy drift by comparing extracted metrics against verified XBRL ground truth data. • User Research: Conduct periodic in-product and out-of-product surveys to assess trust, clarity, and accuracy perception. • Error Categorization & Action: Tag each reported issue as	

Data Extraction, Retrieval Error, Model Hallucination, or UI Miscommunication.	
Start date: Review / End date:	