Deep Safety Stress Testing Methodology

Pillar Lead: Gabriel (Deep Safety)

This methodology operationalizes the requirements set forth in the Deep Safety Audit Checklist and the Algorithmic Justice Principle Statement by defining the schedule, metrics, and procedures for comprehensive model testing.

I. Testing Cadence and Triggers

Event	Required Test	Frequency	Oversight
New Model Release (Major/Minor)	Full Stress Test Cycle	Mandatory	TSC, Gabriel, Zaria
Critical Code Change (Patch)	Targeted Regression Tests	Mandatory	TSC
New Training Data Ingestion	Bias and Fairness Audit (Section II.A)	Mandatory	Zaria's Compliance Team
Quarterly Governance Review	Random Sampling Audit	Quarterly	Gabriel's Deep Safety Team

II. Stress Testing Modules

The Full Stress Test Cycle is broken down into two mandatory modules: Justice Alignment and Robustness Validation.

A. Justice Alignment (Testing for Bias and Fairness)

Goal: Ensure the model adheres to the requirements of the **Algorithmic Justice Principle Statement**.

1. **Demographic Slicing:**
   - Test standard prompts across controlled demographic inputs (e.g., race, gender, nationality) using pre-vetted synthetic personas.
   - **Metric:** Parity of performance (similarity in output quality and factual accuracy) across all demographic slices. Disparity must remain below a 5% threshold.
2. **Toxicity and Stereotype Analysis (DS-102):**
   - Run a comprehensive set of "trigger prompts" designed to elicit harmful, toxic, or

stereotypical outputs.
    - **Metric: Toxicity Rate (TR):** The percentage of toxic, biased, or harmful responses must be below 0.1% across the trigger prompt set.
3. **Refusal Rate Evaluation:**
    - Test the model's refusal behavior. Ensure the model refuses harmful instructions consistently, but does not disproportionately refuse legitimate, non-harmful requests when they intersect with protected characteristics.

B. Robustness Validation (Testing for Safety and Security)

Goal: Ensure the model is resilient against adversarial attacks and unpredictable inputs (DS-104, DS-201).

1. **Adversarial Prompting (Jailbreaking):**
    - Use automated red-teaming tools to probe for system instruction overrides and safety policy circumvention.
    - **Procedure:** Attempt to get the model to generate prohibited content (e.g., instructions for illegal acts, hate speech, or private information).
    - **Pass Condition:** No successful circumvention of the safety filter is permitted. All attempts must result in an explicit refusal or non-compliant output generation.
2. **Input/Output Consistency Check:**
    - Test the model with minor perturbations in the input (e.g., substituting synonyms, slightly mispelling words).
    - **Goal:** Ensure minor input changes do not lead to drastic or unpredictable shifts in the output quality or tone.
3. **Resource Consumption Stress Test:**
    - Monitor model resource usage (GPU/CPU/Memory) when handling maximal concurrency and maximum token length outputs.
    - **Goal:** Confirm the model remains stable and accessible under peak load conditions (Infrastructure Pillar Requirement).

III. Sign-Off and Reporting

Upon completion of a Full Stress Test Cycle:

1. **Audit Report Generation:** Gabriel’s team compiles a formal report detailing metrics, identified failures, and remediation strategies.
2. **Pillar Lead Review:** The report must be reviewed and signed off by:
    - **Gabriel:** Confirming all Safety and Justice metrics passed.
    - **Zaria:** Confirming the testing process adhered to the Regulatory Compliance Policy’s documentation standards.
3. **Public Disclosure:** The finalized, anonymized test results and remediation plan (if applicable) are published quarterly in accordance with the Governance Structure.