**🧠 LAB 4 — AI PROMPT DESIGN FOR RELIABILITY CALCULATION**

**📊 Context**

This lab demonstrates how Generative AI (GAI) can generate synthetic repository data and calculate a **Repository Reliability Index (RRI)** from four key process metrics extracted from GitHub events.  
The RRI is calculated as the **average of normalized metric scores**, representing a simplified AI reasoning approach.

I designed structured prompts to simulate repository performance metrics (X variables) and compute a Repository Reliability Index (Y). The AI model applied a simple averaging formula to quantify repository reliability and assigned qualitative labels. This mirrors how predictive reliability scoring could work in a real system where data is collected automatically from GitHub API events.

**⚙️ Step 1 — Prompt: Generate Mock Metrics (X Variables)**

**Prompt:**

You are an AI assistant analyzing software repository reliability.  
Generate mock values for the following four metrics for 5 GitHub repositories.  
Each metric is on a scale from 0 to 100, where higher is better.  
Metrics:

* issue\_resolution\_time (speed of resolving issues; higher = faster resolution)
* pull\_request\_merge\_rate (percentage of merged PRs)
* commit\_frequency (normalized to 100; higher = more active)
* code\_review\_comments (normalized review depth; higher = more collaborative)  
  Provide results as a JSON array with 5 repositories of mixed performance levels.

**🧮 Step 2 — Prompt: Calculate Reliability Index (Y Output)**

**Prompt:**

Based on the following repository metrics, calculate the *Repository Reliability Index (RRI)* for each repository.  
Formula:

Then classify the reliability based on this scale:

* 0–40 → **Poor**
* 41–60 → **Moderate**
* 61–80 → **Good**
* 81–100 → **Excellent**

Return an Excel output with each repository’s calculated RRI (rounded to 1 decimal) and reliability label.

**📈 Step 3 — Prompt: Summarize Insights**

**Prompt:**

Summarize insights from the computed RRI values.  
Identify which repositories are highly reliable, which are risky, and provide one short suggestion per repository on how to improve reliability (e.g., increase review activity, resolve issues faster, etc.).

**🧠 Step 4 — Prompt: Final AI-driven Interpretation**

**Prompt:**

Based on the RRI results, generate a short paragraph (4–5 sentences) that an AI analyst could include in a reliability assessment report.  
Mention which repositories show strong software process maturity and which indicate risk areas.