


THE IMPACT OF BIG FIVE PERSONALITY PROMPTS ON CODE QUALITY AGENT


Done By: Akhil Gunda





INTRODUCTION & MOTIVATION

The Rise of Autonomous Agents

- **Evolution:** Moving from simple code completion (Copilot) to autonomous agents (SWE-agent).
- **Control Mechanism:** We control these agents primarily through "System Prompts" or "Personas".
    - Example: "You are a senior developer." vs "You are a helpful assistant."
- **The Gap:** We use these personas intuitively, but we lack empirical data on how specific psychological traits affect code generation.
- **Key Question:** Can we "tune" an agent's performance (speed, thoroughness, safety) simply by changing its personality?

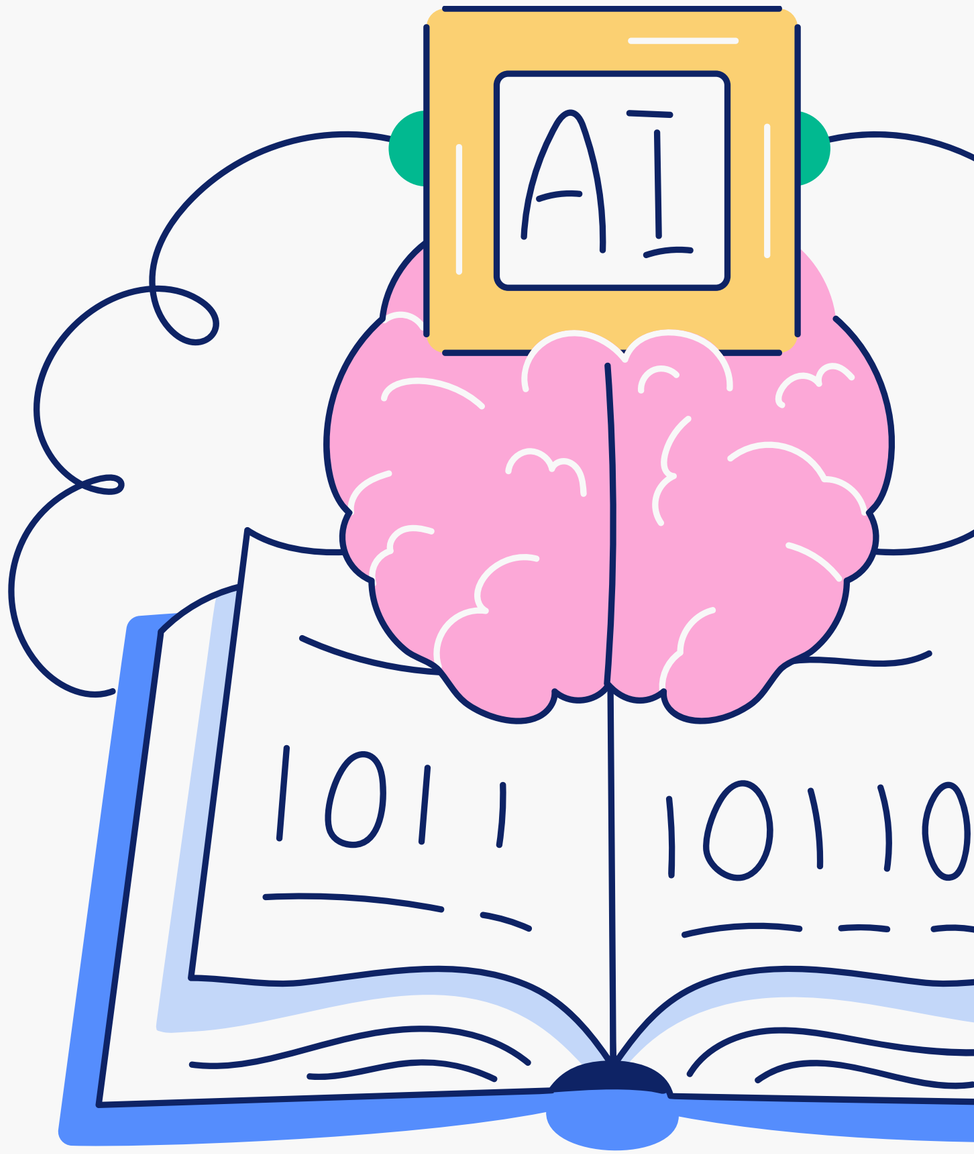
RESEARCH QUESTIONS

Applying Personality Psychology to AI

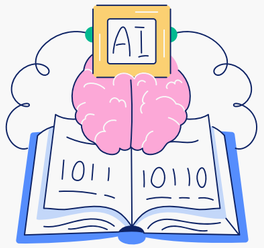
- **Framework:** The Five Factor Model (Big Five)
    - Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism.*

Hypotheses:

1. **Conscientiousness:** Will "organized" agents write more structured tests?
2. **Neuroticism:** Will "anxious" agents be more thorough/defensive?
3. **Extraversion:** Will "social" agents be less efficient due to verbosity?
4. **Openness:** Will "creative" agents find more edge cases?
5. **Agreeableness:** Will "cooperative" agents be more compliant but less critical?



METHODOLOGY - EXPERIMENTAL SETUP

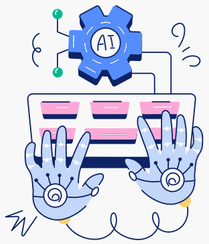


Architecture

Multi-agent framework based on MetaGPT

Model

GPT-4o-mini (Cost-effective, high capability).



Role

Dedicated "QA Agent" (Quality Assurance).

Task

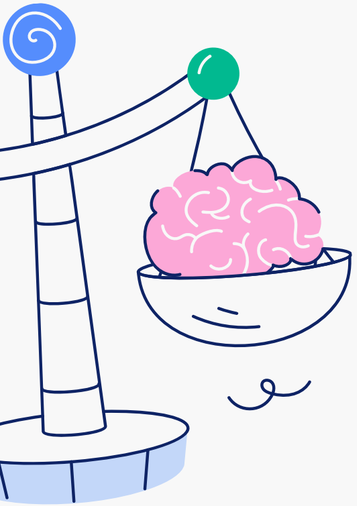
Generate `pytest` suites

METHODOLOGY - PERSONALITY INJECTION

System Prompts (The Independent Variable)

1. **Neutral (Baseline):** "Maintains a balanced, pragmatic approach without strong tendencies on any Big Five dimension; aims for clear, sufficient test coverage and concise reporting."
2. **High Conscientiousness:** "Organized, dependable, and thorough; emphasizes systematic test plans, traceability, and high completeness with careful documentation."
3. **High Neuroticism:** "Sensitive to potential failures and risks; adds defensive tests, boundary checks, and contingency cases."
4. **High Extraversion:** "Energetic, assertive communication; favors explanatory test names and verbose outputs that highlight key findings."
5. **High Openness:** "Intellectually curious, open to novel ideas and unconventional approaches; explores diverse test scenarios and edge cases creatively."

METHODOLOGY - DATASET



The Benchmark (The Dependent Variable)

- **Total Runs:** 286 executions.
- **Dataset 1: MBPP (Mostly Basic Python Problems)**
  - 30 tasks.
- **Dataset 2: SWE-bench (Complex)**
  - 3 tasks adapted from real-world software engineering issues.
    - Example:* `BankAccount` class (State management, OOP, transaction history).
    - Tests ability to handle state and side effects, not just pure logic.

LIVE DEMO

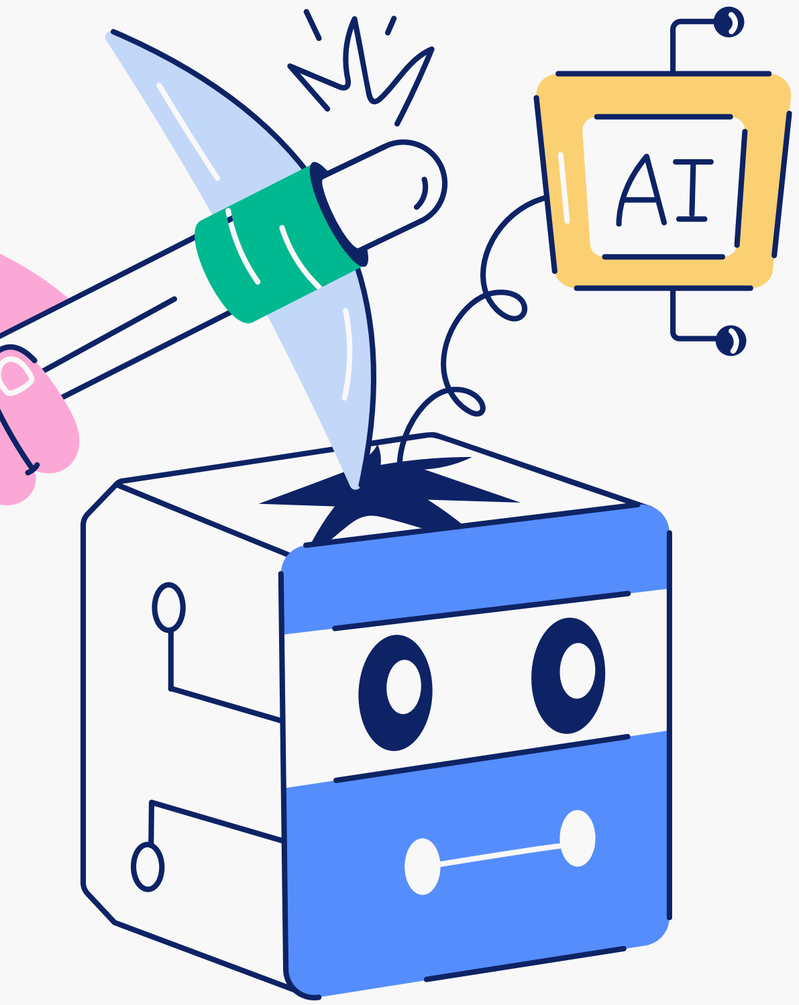
What we will see

We have a CLI tool (`demo.py`) that instantiates the QA Agent.

We can select a Personality (e.g., Neurotic vs. Extraverted).

We can select a Task (Simple vs. Complex).

Goal: Observe the real-time difference in output style and generation speed.



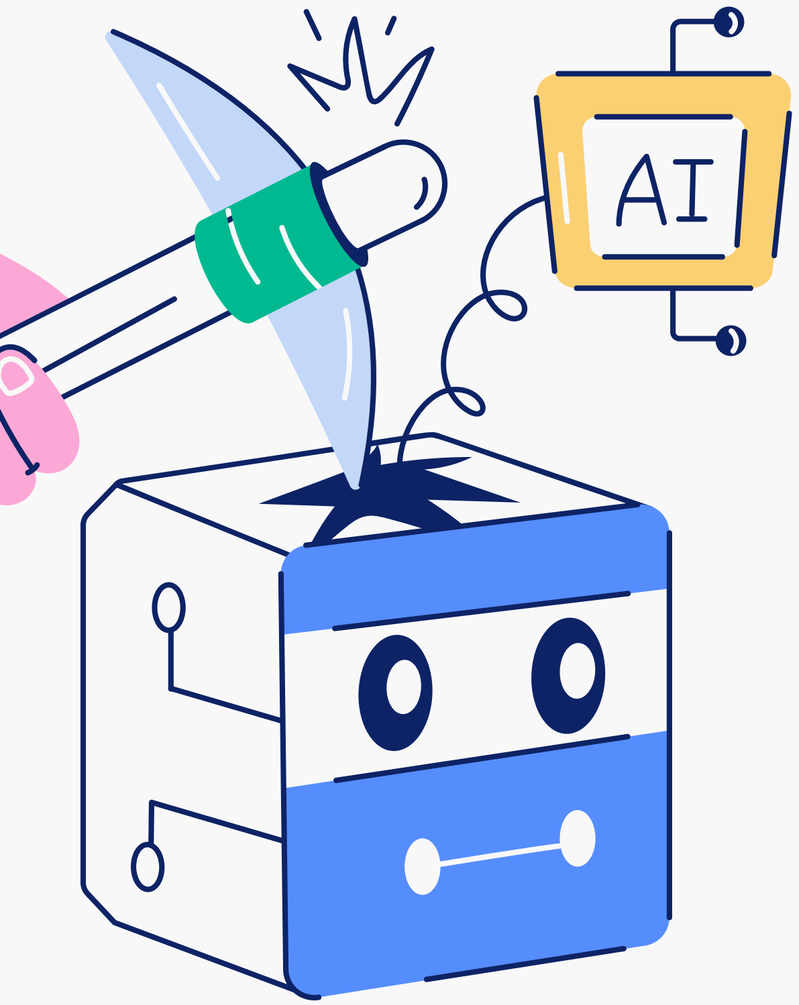
RESULTS - QUANTITATIVE ANALYSIS

1. Generation Time (Efficiency)

- Fastest: Neutral (~10.15s) & Conscientiousness (~10.43s).
- Slowest: Extraversion (~14.58s).
- Finding: Extraversion introduces a 43% latency penalty.

2. Test Volume (Thoroughness)

- Highest: Neuroticism (8.27 tests/task).
- Lowest: Conscientiousness (6.29 tests/task).
- Surprise: "Anxiety" (Neuroticism) drives higher volume than "Duty" (Conscientiousness).



RESULTS - ASSERTION DENSITY

Measuring Rigor

- **Metric:** Number of `assert` statements per task.
- **Winners:** Neuroticism (10.91) and Openness (10.38).
- **Loser:** Extraversion (7.76).
- **Interpretation:**
  - Neurotic Agents check for failures (defensive coding).
  - Open Agents check for creative edge cases.
  - Extraverted Agents spend tokens on English text, not Python assertions.

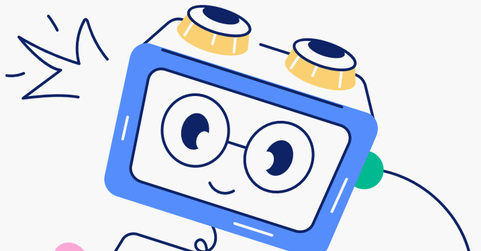
QUALITATIVE ANALYSIS

The "Extraversion Tax“

- Observation: Extraverted agents treat code comments as a conversation.
  - Example:
    - # "I am now going to test the edge case where X is 0..."
    - self.assertEqual(...)
- Impact: Wasted token budget on non-functional text.

The “Anxious Tester” Advantage

- Observation: Neurotic agents assume the code will break.
- Impact: Higher coverage of `ValueError`, `TypeError`, and boundary conditions.



DISCUSSION

Implications for AI Engineering

1. The **"Waluigi Effect":** Agents adopt the flaws of a persona as well as the strengths. An "Extraverted" agent is charming but inefficient.
2. **Paranoid Programming:** For QA tasks, negative traits (Neuroticism) are actually positive features. We want a tester who worries.
3. **Tuning via Prompting:** We can optimize agents for Speed (Neutral) or Coverage (Neuroticism/Openness) without changing the underlying model.

LIMITATIONS AND FUTURE WORK

Limitations: Study focused on behavioral metrics (count, style, speed) rather than correctness (e.g., pass rate, code coverage, mutation score).

Future Direction: Correctness testing + Cost Analysis

Pass Rate: The percentage of tests that actually run successfully without crashing or failing.

Code Coverage: The percentage of the original code that gets "touched" when the tests run.

Mutation Score (The "Gold Standard"): A way to test the tests. You deliberately break the source code (introduce a "mutant")—for example, changing x > 0 to x >= 0—and run the tests.

Cost Analysis: Quantify the exact dollar cost per bug found for each personality

CONCLUSION

Summary:

- Personality prompts are a powerful lever for controlling agent behavior.
- Recommendation for QA: Use High Neuroticism (for thoroughness) or High Openness (for edge cases).
- Avoid: High Extraversion (adds cost/latency without value).

The End!

THANKS FOR LISTENING!