

StressSpec

Requirements Stress Tester

Individual Project – Jeffrey Perdue

The Problem

- 37% of enterprise project failures come from **poor requirements**.
- Fixing requirement defects late costs **5–10x more**.
- Current tools clarify requirements but **don't stress-test for risks**.
- Teams discover ambiguity, conflicts, and gaps **after coding begins**.

Why It Matters

- **Save time & money** by catching issues early.
- **Improve quality** with testable, realistic, compliant requirements.
- **Support collaboration** via traceable, prioritized risk reports.
- **Industry relevance:** showcases AI + rule-based analysis in SE.

The Solution

Python-based “Wind Tunnel” for Requirements

- Input Ingestion (.txt/.md)
- Parsing & Labeling (IDs + line numbers)
- Risk Detection (ambiguity, security, conflicts, etc.)
- Configurable Rules (rules.json, domain profiles)
- Severity Scoring (High/Med/Low + Top 5 risks)
- Multi-format Reporting (Markdown, CSV, JSON, HTML)

Sprint 1 (MVP – 4 Weeks)

- **Input Ingestion** – load .txt/.md
- **Parsing & Labeling** – assign IDs + traceability
- **Risk Detection (basic)** – ambiguity, availability, performance, security
- **Configurable Rules** – rules.json support
- **Reporting** – Markdown, CSV, JSON
- **Severity Scoring** – basic High/Medium/Low

Deliverable: End-to-end MVP flow

Sprint 2 (Expansion – 5 Weeks)

- Risk Detection (expanded) – add privacy, scope, traceability, conflicts
- Severity Scoring (expanded) – totals + Top 5 riskiest
- Reporting (enhanced) – summaries, HTML/visuals
- Configurable Rules (extended) – multiple domain profiles

Deliverable: Polished tool with 8 categories + stretch visualization

Summary

StressSpec = A requirements stress-tester

- Catches risks before coding begins
- Saves cost, improves quality, builds collaboration
- Delivers a working, configurable tool in **two sprints**

