StressSpec Week 3 Progress Report

Requirements Stress Tester - Risk Detection & Reporting Complete

Individual Project – Jeffrey Perdue

Week 3: Detectors + Reporting + CLI Enhancements

**

✅ Week 3 Highlights

- Implemented multi-format reporting (Markdown, CSV, JSON)
- Added Performance & Availability detectors
- Enabled configuration via `data/rules.json`
- Integrated detectors through Factory pattern
- Added analyzer pipeline and end-to-end reporting

📦 New Modules Delivered

```
src/
├── analyzer.py                 # Runs detectors and aggregates risks
├── reporting/
│   ├── base.py                 # ReportFormat, ReportData, Reporter interface
│   ├── markdown_reporter.py    # MD report writer
│   ├── csv_reporter.py         # CSV report writer
│   └── json_reporter.py        # JSON report writer
└── detectors/
    ├── performance_detector.py # Performance risk detection
    └── availability_detector.py# Availability risk detection
```

🧭 Architecture Updates

- Factory Method: `RiskDetectorFactory` now registers 6 detectors
- Strategy + Template Method: shared detection workflow in `BaseRiskDetector`
- Config-Driven Rules: `data/rules.json` toggles detectors, sets severities
- Analyzer: `analyzer.py` coordinates detection and grouping

🧪 Testing Results (Week 3)

- Tests executed: `python -m pytest -q tests`
- Status: 31/31 tests passing (unit + integration)
- New unit tests: performance, availability
- Integration: verifies end-to-end pipeline and risks-by-requirement

🎯 Key Metrics & Numbers

- Detectors implemented: 6 (Ambiguity, Missing Detail, Security, Conflict, Performance, Availability)
- Report formats: 3 (Markdown, CSV, JSON)
- Tests: 31 passing (unit + integration)
- Core modules added this week: 5 (`analyzer.py` , 3 reporters, 2 detectors)
- Rules: Config-driven severities and enable flags in `data/rules.json`

🖨️ Reporting Examples

- Markdown: `report.md` with per-requirement sections and summary
- CSV: row-per-risk for spreadsheet analysis
- JSON: machine-readable structure with full risk objects

Run:

```
python main.py --file data/sample_requirements.txt --report-format md --verbose
```

📊 Current Coverage of Risk Types

- Ambiguity ✅
- Missing Detail ✅
- Security ✅
- Conflict ✅
- Performance ✅
- Availability ✅

🧩 CLI Enhancements

- `--report-format md|csv|json`
- `--output <path>`
- Verbose run output

📌 Summary

- Sprint 1 features: Well on-track ✅
- Robust reporting and detectors in place
- Configurable, test-backed, and extensible architecture

Quick Commands

```
python main.py --file data/sample_requirements.txt --report-format md
python main.py --file data/sample_requirements.txt --report-format csv --output out.csv
python main.py --file data/sample_requirements.txt --report-format json --output out.json
python -m pytest -q tests
```