

Developer-facing Report: Sales Analysis Prototype

Alexander Ricciardi

Colorado State University Global

CSC505: Principles of Programming

Professor: Dr. Joseph Issa

November 16, 2025

Developer-facing Report: Sales Analysis Prototype

This developer-facing report answers questions from stakeholders about the Sales Analysis prototype questions. Questions about the script's purpose, libraries or tools choices, challenges, planned improvements, and lessons learned related to software development practice.

Stakeholder-Facing Questions

1. What was the purpose or intended use case of your script?

The purpose of the script is to prototype a sales analysis tool for Omega.py Online Store. The prototype is a console-based tool that loads a monthly CSV file of sales data and computes metrics such as total revenue and total items sold, order statistics such as minimum and mean, and statistics such as revenue by product, category, and date. The prototype provides the developing team with a quick way to test the tool logic and allows showcasing the tool functionality to stakeholders, ensuring that it aligns with their requirements.

2. What tools or libraries did you use, and why?

To handle CSV sales data tabulation and revenue aggregation computation, the prototype leverages the Panda library, as the library provides reliable data CSV statistical operations, such as category grouping and aggregation. To handle basic sales metric computations, the script uses the NumPy library, which provides very efficient numerical-based arrays and statistical computations, as the library is written in the C programming language. The prototype uses the Matplotlib library to visualize a daily revenue line graph, as the library is a standard plotting tool in Python that generates visual charts, and to render console UI, the script uses the internal utilities package, as it encapsulates easy-to-use ASCII-based banners, menus, and input validation.

3. What challenges did you encounter during development?

The main challenge encountered was to ensure that the program logic is modular, that is, to properly separate from the console UI loop the file CSV data-loading process, tabulation, and the analysis logic, so the code can be more readable and testable. Another challenge was formatting and parsing the order dates and currency fields properly to ensure that the statistical calculations were computed correctly.

4. How would you expand or improve this prototype in future iterations?

In future iterations, the development team should implement schema validation, which involves verifying that the loaded CSV schema columns and their values, such as dates and numeric fields, are formatted and parsed correctly. Additionally, the team should implement functionality that handles cases such as missing file(s) and empty datasets. Also, test units and an event logger should be implemented to automatically test the program logic and generate an error event log to help the debugging and troubleshooting process.

5. What lessons did you learn that apply to broader software development work?

Through the process of prototyping a feature by first coding a light version of it, it was recognized that this process is an efficient method to validate requirements with stakeholders and reduce the risk of over-engineering the wrong implementation. It was also recognized that this method integrates well with the Agile methodology, more specifically the sprint concept, which breaks down a larger problem or system into smaller, incremental, and more manageable iteration pieces (e.g., the prototype script), helping software development teams to make measurable and safer design and implementation decisions that align with stakeholders' needs and requirements.