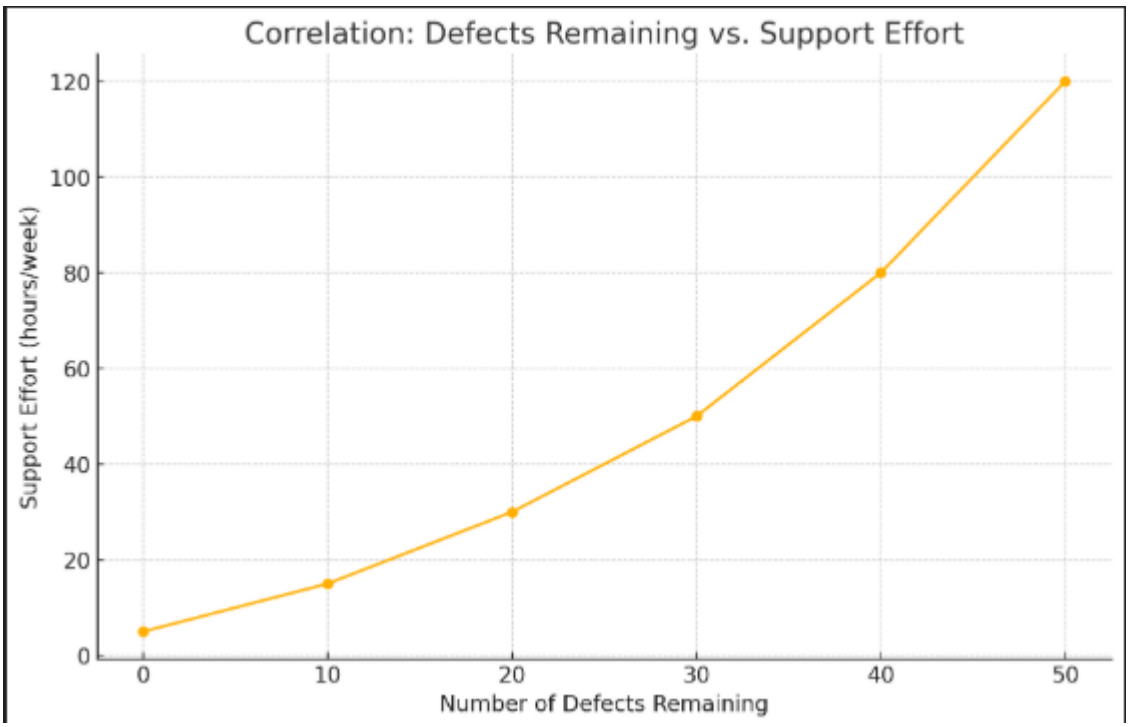
Exercise 9

1.

When software ships with defects still open, product-support burden rises—often nonlinearly. Fewer defects mean:

Lower call-volume and ticket count
Faster resolution times (issues tend to be simpler/isolated)
Less developer involvement in triage and bug-fixing
Higher customer satisfaction and lower escalation rates

Conversely, more defects remaining drive up support in terms of hours, personnel, and cost, as illustrated below.



2.
for:

Deep Domain Knowledge
Developers know the requirements and design intimately, so they can target edge cases that generic testers might miss.

Rapid Feedback Loop
Immediate unit and integration tests speed development, catching regressions on the spot.

Cost-Efficiency
Early defect detection by devs reduces hand-offs to QA and the associated administrative overhead.

Ownership & Accountability
When devs write tests, they take greater responsibility for code quality and are motivated to avoid sloppy work.

Sharpened Design
Writing tests (especially in TDD) forces clearer interfaces and modular design, improving maintainability.

Against:

Confirmation Bias
Devs may subconsciously avoid testing “pathological” cases they assume won’t happen.

Limited Perspective
They may overlook usability issues or real-world workflows that non-developer testers would catch.

Over-Optimization
Focus on “happy path” scenarios rather than robustness under load or error conditions.

Skillset Mismatch
Effective testing (performance, security, accessibility) often requires specialized tools and expertise.

Time Pressure
Under tight deadlines, devs may skimp on comprehensive testing in favor of shipping features.

4.
Regression testing is the process of re-running a set of previously executed tests after changes (bug fixes, enhancements) to confirm that existing functionality still works as intended

5.
Black-Box Testing

Tester knows *what* the software should do but not *how*.

Focuses on inputs, outputs, and overall behavior (functional testing, user acceptance tests).

Pros: Unbiased by implementation; catches missing requirements.

White-Box Testing

Tester has full knowledge of internal code structure and logic.

Involves unit tests, path/branch coverage, code-walkthroughs, and static analysis.

Pros: Can achieve high coverage; finds internal defects early.