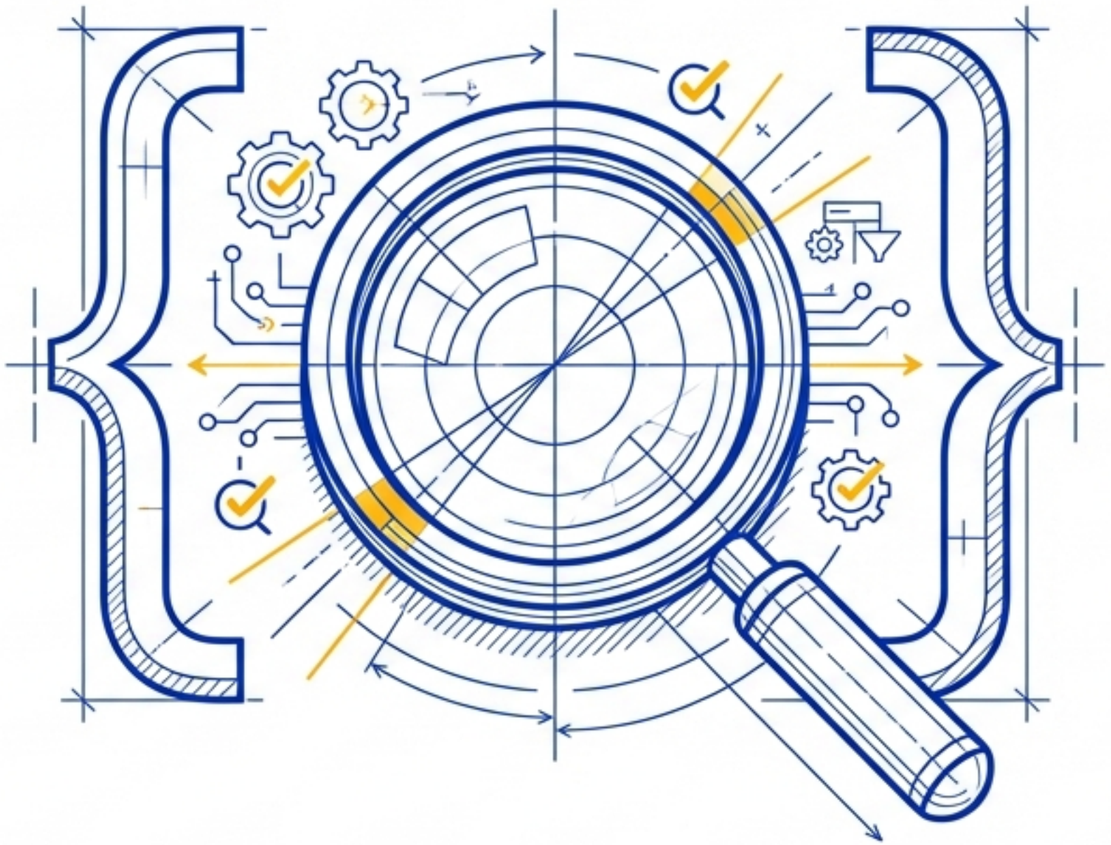
Continuity of Quality: The Shift-Left Mindset

A Strategic Guide to Testing Across the SDLC

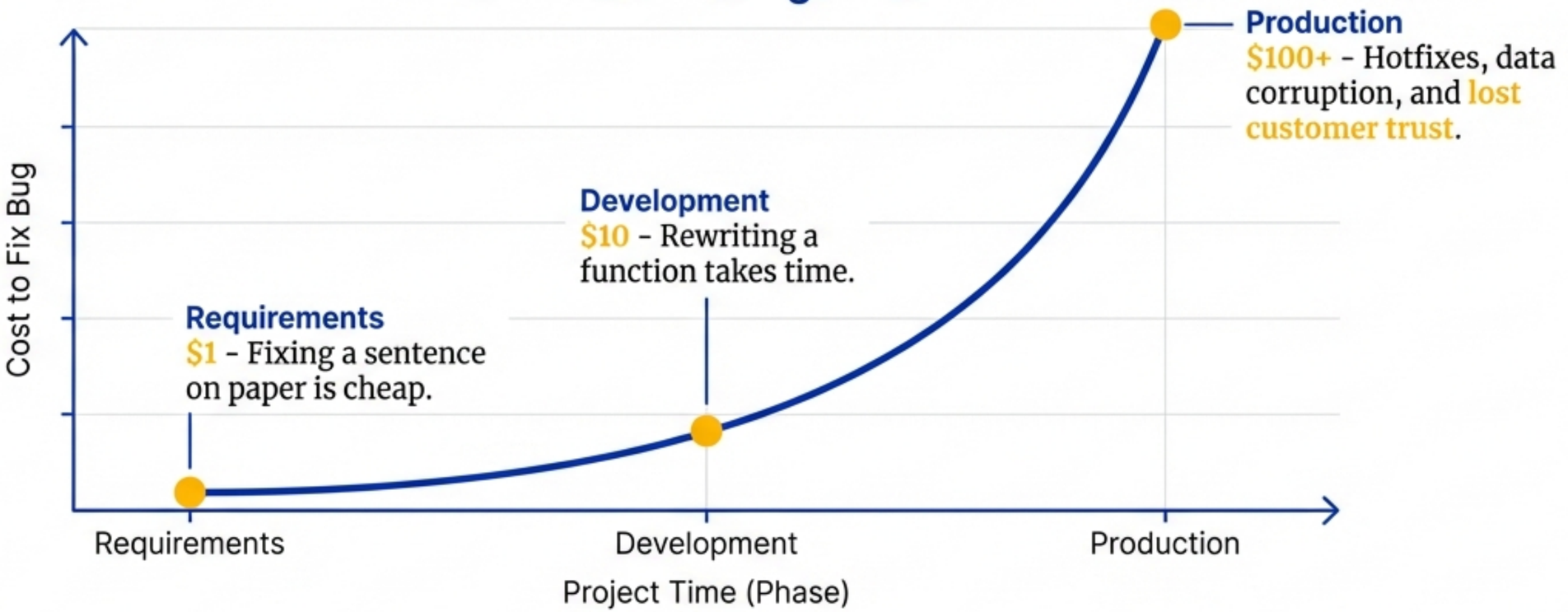


Learning Material from Skill-Wanderer (Non-Profit Ed-Tech)

The Golden Rule: The Cost of Quality

Testing is an activity that happens continuously, not a phase that happens at the end.

The Cost of Change Curve

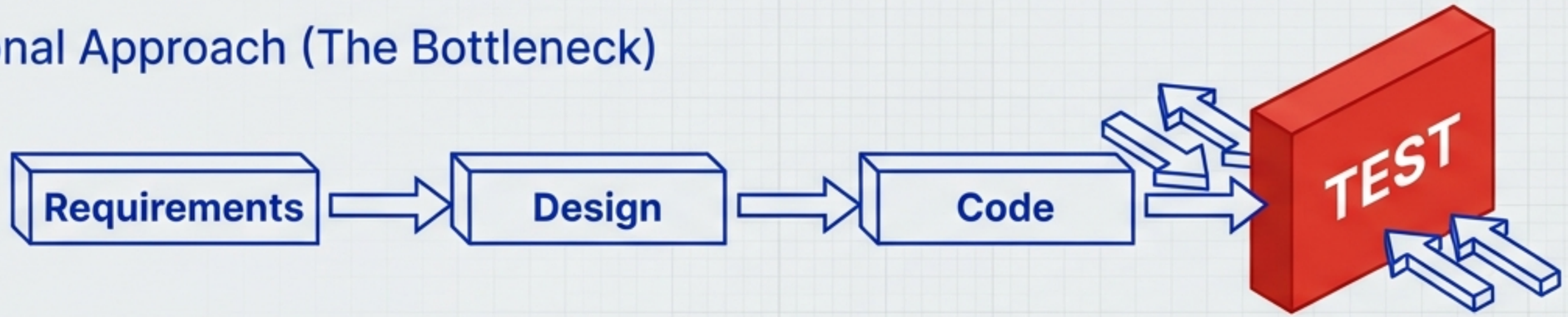


Pro Tip

Your job as a tester isn't just to find bugs in the software; it's to find bugs in the IDEAS before they become software.

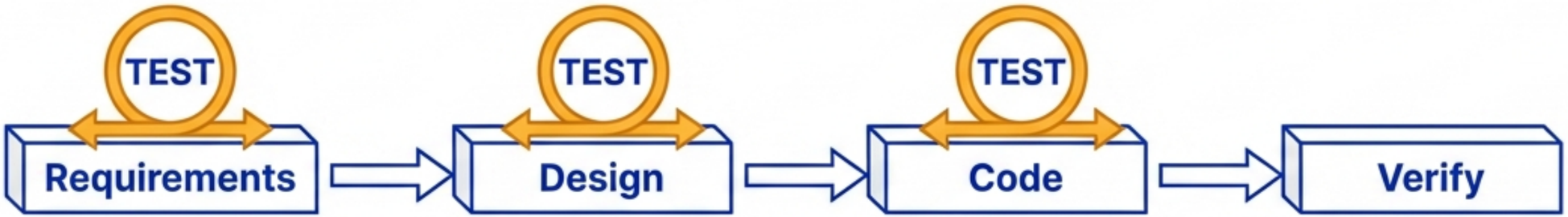
Moving from “Bottleneck” to “Continuous Quality”

Traditional Approach (The Bottleneck)



If you wait until the code is finished to start testing, you are too late.

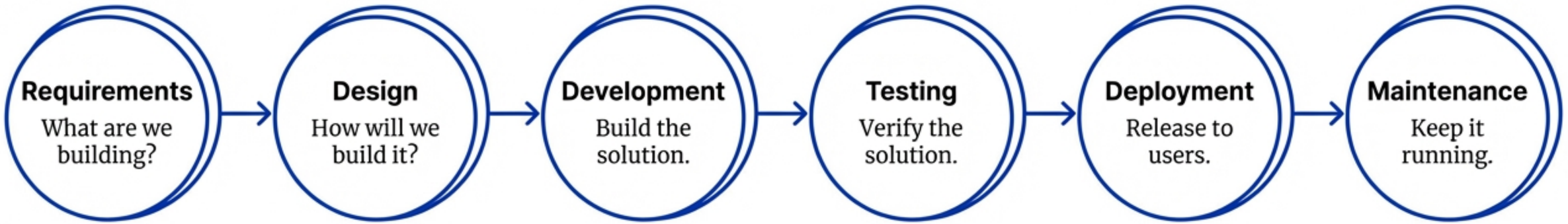
Shift-Left Approach (Continuous Quality)



Testing activities move ‘Left’ (earlier) on the timeline. Quality is built in, not inspected in.

The SDLC Ecosystem

We will now explore your specific responsibilities in each of these phases.



Phase 1: Requirements Analysis (Testing Ideas)

The Tester's Role: Static Testing

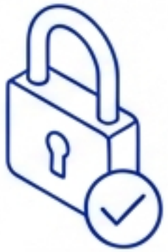
You are reviewing documentation, not code. Your goal is to hunt for ambiguity. Ask:

"What happens if the user loses internet here?" or "What determines success?"

❌ Bad Requirement

- "The page should load fast."
- **Vague, untestable.**
- "The password should be strong."

✅ Testable Requirement

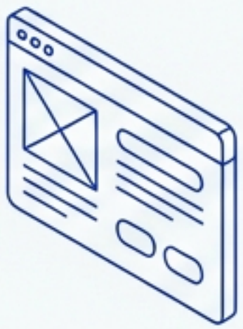


- "The page must load in under 2 seconds with a 4G connection."
- **Specific, measurable.**
- "The password must have 8+ characters, 1 number, and 1 symbol."

Phase 2: Design (The Blueprint Check)

Ensuring the technical design supports business needs.

Design Review



Scrutinize wireframes and UI designs. Be the voice of the user before a pixel is drawn.

Data Flow

Check logic. If a user saves data here, does it logically show up there?

Risk Analysis

Identify where the feature is most likely to break.

Critical Question

Apply "What If" scenarios to the blueprint. Example: If the design shows an "Upload Photo" button, ask: What is the max file size? What format? What happens if the file is corrupt?

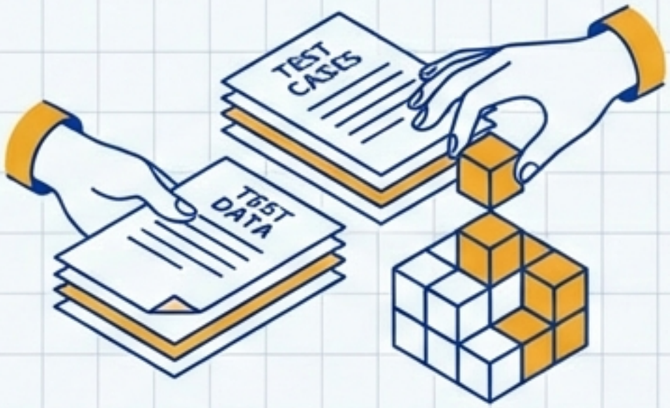
Phase 3: Development (Preparation, Not Downtime)

The Developer

Building the software solution.

The Tester

Preparing the Safety Net.



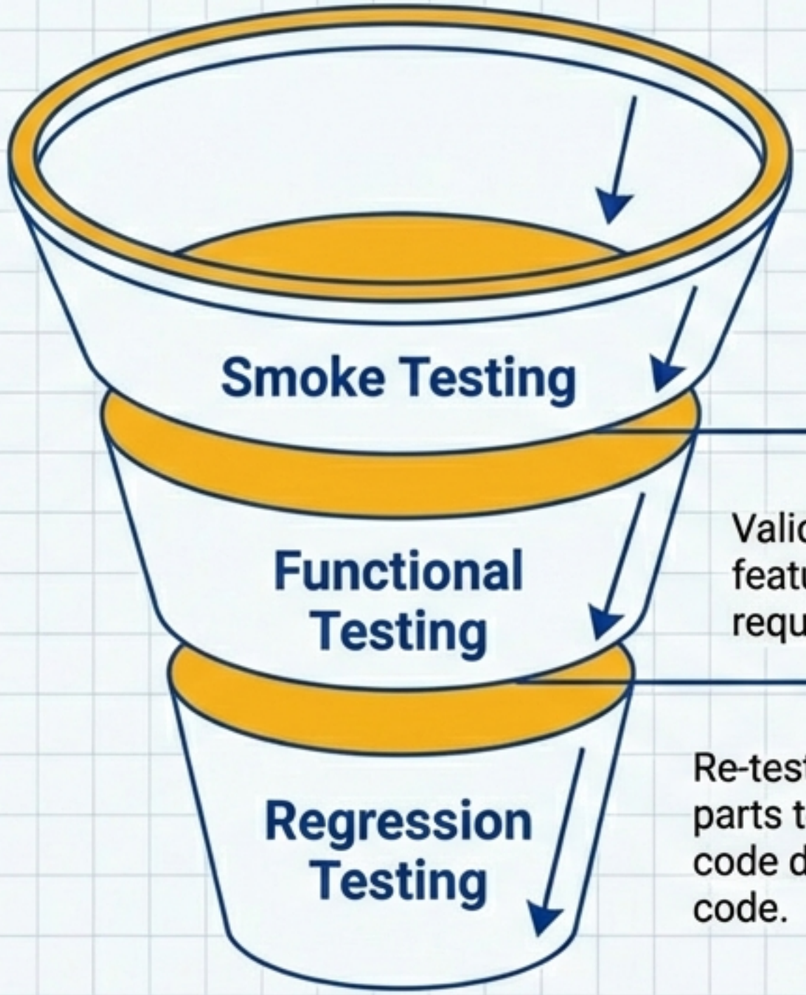
- **Write Test Cases:** Create step-by-step instructions to run later.
- **Prepare Test Data:** Generate dummy users, fake credit card numbers, or sample files.
- **Traceability Matrix:** Create a link between Requirements and Tests to ensure nothing is missed.

While developers write code, you are preparing the instruments for validation.

Phase 4: The Execution Phase

Entry Criteria

- Code is frozen and deployed to QA environment.



The "Health Check". Does the app launch? If no, reject build.

Validating specific features against requirements.

Re-testing unchanged parts to ensure new code didn't break old code.

Exit Criteria

- No critical bugs (Sev 1 or Sev 2) remain.

Phase 5: Deployment (The Safety Net)

Releasing safely to the customer.

The Actions

Sanity Testing

A quick check in the Staging or Production environment.

Release Verification

Confirming the correct version is live.

⚠ CRITICAL WARNING

Do Not Perform Destructive Testing in Production.

Destructive testing involves intentional failure points (e.g., deleting databases to test recovery). This must never happen in the live environment.

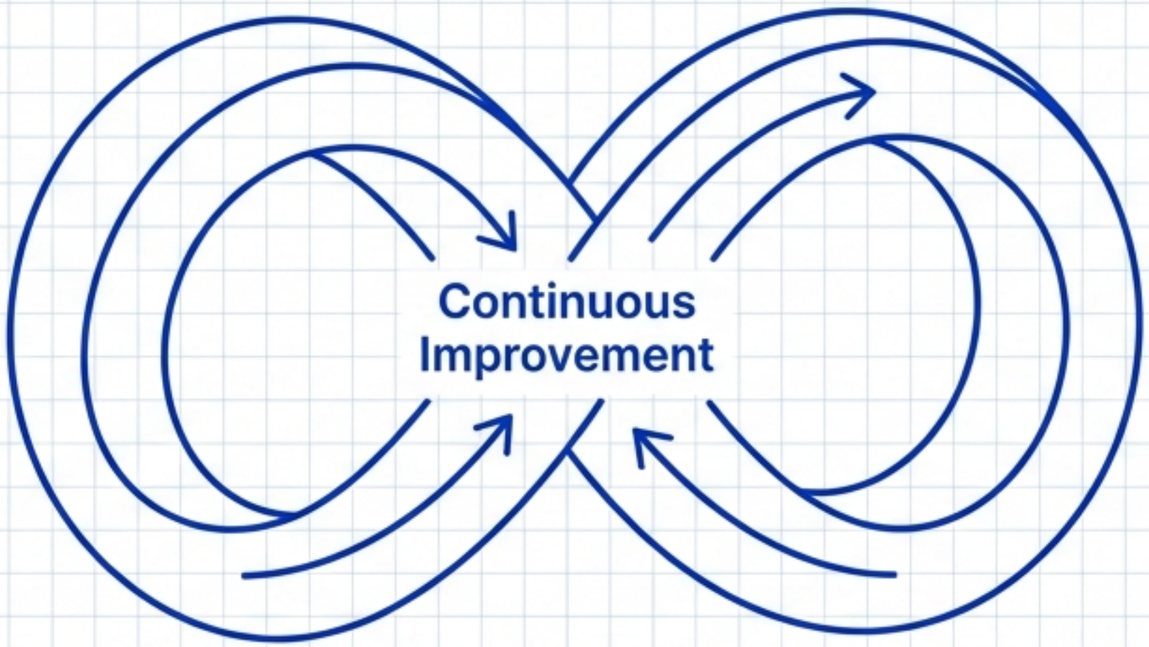
Ensure a smooth and safe transition to the live environment with final verification and strict adherence to safety protocols.

Phase 6: Maintenance (Evolution)

Software is never truly ‘done’.

Patch Testing

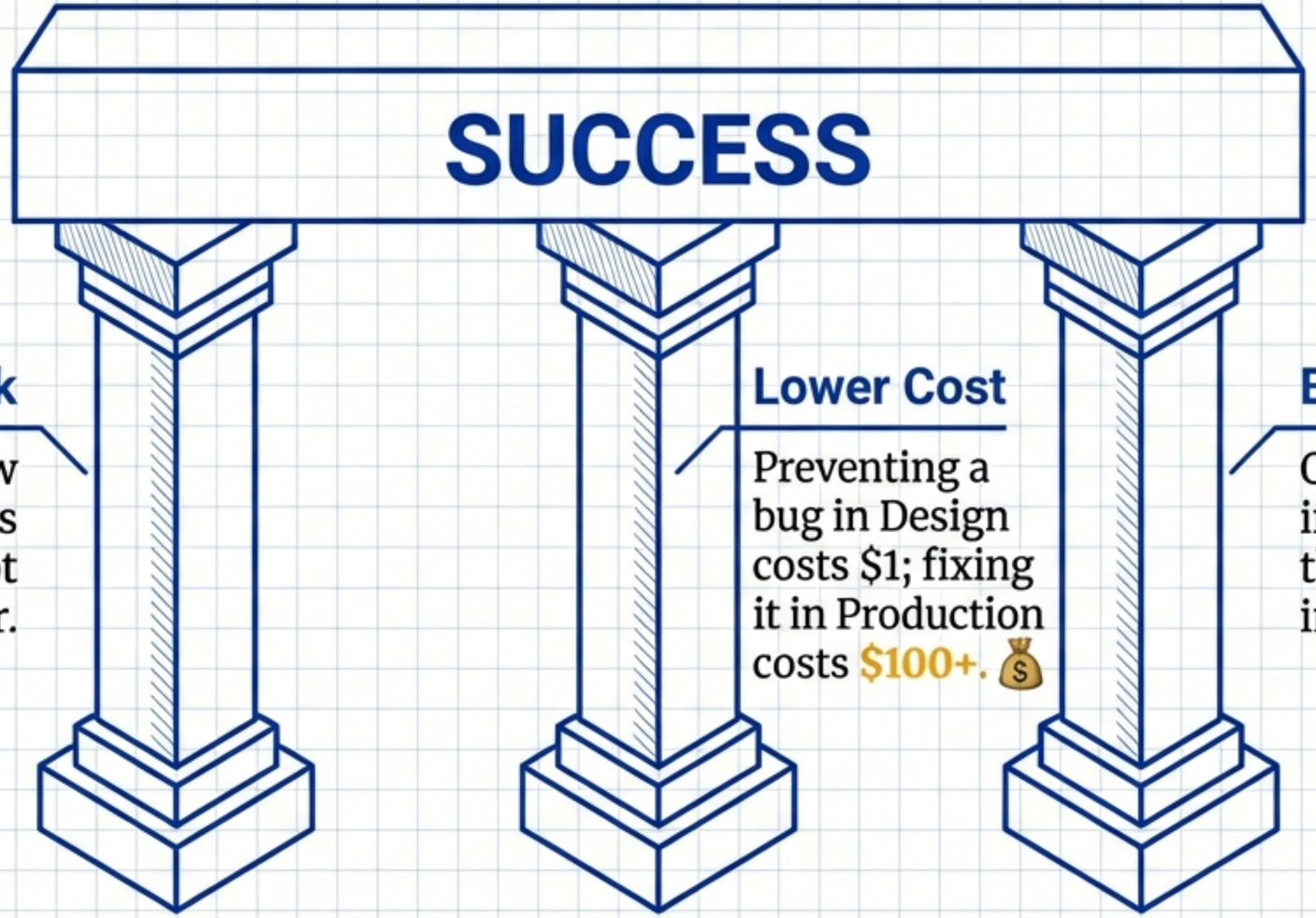
When a fix is applied, execute Regression Testing on surrounding areas to ensure stability.



Incident Analysis

When users report bugs, your job is to reproduce them in the test environment to help developers fix them.

The Impact of Shift-Left



Faster Feedback

Developers know about bugs immediately, not weeks later.

Lower Cost

Preventing a bug in Design costs $1; fixing it in Production costs $100+. 💰

Better Quality

Quality is designed into the product from the start, not just inspected at the end.

Your Quality Checklist

You are now ready to apply the Shift-Left mindset.

- [x] Explain why testing must happen continuously.
- [x] Identify responsibilities in Requirements, Design, and Coding.
- [x] Explain the “Cost of Quality” (highlighted in Warm Amber: $1 vs $100).
- [x] Apply early testing strategies to daily work.

Skill-Wanderer. Empowering your learning journey.