**User Story Ambiguity & Risk Analysis**

**Project:** *Demo Blaze E-commerce Testing***Prepared By:** *Maysoon Helmy*   
**Date:** *26 September 2025*

During the review of the initial user stories for the Demo Blaze platform, several areas were identified where requirements lack specificity, contain assumptions, or omit edge cases. While the current stories capture core user intentions, refining them will significantly improve testability, reduce ambiguity during development, and prevent defects caused by misinterpretation.

Below are ten key issues observed across the three user stories, along with actionable recommendations for improvement.

**1. Subjective Language in Benefit Statement**

Issue: The phrase “find relevant products quickly” includes the term “quickly,” which is subjective and not measurable. Performance expectations should be quantifiable or omitted if not part of functional scope.  
Recommendation: Remove “quickly” or replace it with a defined performance criterion (e.g., “within 2 seconds of selection”).

**2. Undefined User Authentication State**

Issue: The user stories refer to “a shopper” but do not specify whether the user is a guest or logged in. Demo Blaze supports guest browsing and cart management, but this distinction affects test coverage and system behavior.  
Recommendation: Clarify user state by stating: “As a guest or registered shopper…” in relevant stories.

**3. Vague Scope of Cart Management**

Issue: The phrase “manage my shopping cart” is too broad. It could imply adding, removing, editing quantities, or applying discounts—none of which are explicitly defined.  
Recommendation: Narrow the scope to specific actions: “view items, remove products, and proceed to check out.

**4. Missing Error or Empty-State Handling**

Issue: Acceptance criteria only describe successful paths (e.g., clicking “Phones” displays phones). There is no coverage for scenarios where a category is empty or fails to load.  
Recommendation: Add a negative test condition: “Given the ‘Monitors’ category contains no products, when the user selects it, then display ‘No products available.’”

**5. Assumption That All Categories Are Always Present**

Issue: The stories assume that “Phones,” “Laptops,” and “Monitors” will always appear in the UI. However, if backend data changes or a category is temporarily disabled, the frontend should handle this gracefully.  
Recommendation: Include a validation requirement: “The category navigation must reflect only active product categories from the backend.”

**6. Undefined Pagination Behavior**

Issue: The UI includes “Previous” and “Next” buttons, but the number of products per page and behavior on the last/first page are not specified.  
Recommendation: Define pagination rules: “Each page displays up to 9 products. The ‘Next’ button is disabled on the last page; ‘Previous’ is disabled on the first.”

**7. Insufficient Cart Feedback After Add-to-Cart**

Issue: The only confirmation after adding an item is a transient JavaScript alert. There is no persistent visual indicator (e.g., cart icon badge) to confirm the action.  
Recommendation: Enhance user feedback: “After adding an item, the cart icon in the navigation bar must display the updated total item count.”

**8. No Handling for Duplicate Product Additions**

Issue: It is unclear whether adding the same product multiple times creates duplicate entries or increments the quantity of a single entry. This affects cart accuracy and checkout totals.  
Recommendation: Specify behavior: “Adding a product already in the cart increases its quantity by 1 instead of creating a new line item.”

**9. Lack of Input Validation in Checkout Flow**

Issue: The “Place Order” form accepts any input without validation (e.g., empty name, invalid credit card format). This could lead to failed orders or poor user experience.  
Recommendation: Add validation rules: “If the ‘Name’ field is empty upon clicking ‘Purchase,’ display ‘Please fill out Name’ and prevent form submission.”

**10. No Consideration for Responsive or Mobile Use**

Issue: All user stories and acceptance criteria are written from a desktop perspective. Demo Blaze is accessed on mobile devices, where layout, tap targets, and navigation may behave differently.  
Recommendation: Include cross-device compatibility: “All described user flows must function correctly on standard mobile and tablet viewports.”

**Conclusion**

Addressing these points will strengthen the foundation of the test design and reduce the risk of requirement gaps leading to defects. The suggested improvements align with best practices in testable requirement writing and support end-to-end traceability from user need to test validation.