BA TOOLKIT · FREE TEMPLATE

Acceptance criteria template

Five checks, vague-word swaps and Given/When/Then criteria a tester can't misread, with a filled example.

By **Prachit Parikh**, business analyst · github.com/prachitparikh/ba-toolkit

Template from the BA Toolkit by Prachit Parikh. Replace everything in [brackets]. Walkthrough video: How to Write Acceptance Criteria Testers Can't Misread.

Item	Detail
Story	[NS-142: Refund requests]
Story owner	[Product owner name]
BA	[Your name]
Version / date	[v0.1, YYYY-MM-DD]
Status	[Draft / In review / Agreed]

1. Run the five checks before the story goes to the team

- [ ] **No vague words.** Search the criteria for: quickly, easily, user-friendly, as needed, should support, fast, intuitive, appropriate, etc. Swap each one for a number or a named behavior.
- [ ] **One behavior per criterion.** Watch for "and" in the outcome. Two behaviors means two criteria, so each one can pass or fail on its own.
- [ ] **Given, When, Then with real data.** A real starting state, one action, an observable result. A tester should be able to set it up without calling you.
- [ ] **The unhappy path.** For each happy-path criterion, ask: what if the input is too big, too late, a duplicate, or missing? Write those down too.
- [ ] **The stranger test.** Hand the criteria to someone who wasn't in the meetings. If they can't write a test for it in about two minutes, it isn't done yet.

2. Vague word swaps

Vague	Testable replacement	Agreed with
[quickly]	[within 2 business days]	[Finance ops lead]

Vague	Testable replacement	Agreed with
[user-friendly]	[the customer can submit a refund in 3 steps or fewer]	[UX lead]

3. Criteria

Write the outcome as something a person can see: a status, a message, an amount, a record. Not "the database gets updated." Describe behavior, not buttons ("click the blue button" says how, not what should happen).

ID	Given (starting state)	When (one action)	Then (observable result)	Path	Test case
AC-01	[An order paid by card 20 days ago]	[The customer requests a full refund]	[The refund shows as Approved for the exact amount paid]	Happy	[TC-011]
AC-02	[A refund was approved]	[The approval is saved]	[The customer gets a confirmation email within 10 minutes]	Happy	[TC-012]
AC-03	[An order paid $120]	[The customer requests a $150 refund]	[The request is blocked and they see "Refund can't be more than the amount paid."]	Unhappy	[TC-013]
AC-04	[An order that was already refunded]	[The customer asks for another refund]	[The request is blocked and they see "This order has already been refunded."]	Unhappy	[TC-014]
AC-05	[An order placed 91 days ago]	[The customer requests a refund]	[The request is blocked and they see the 90-day policy message]	Unhappy	[TC-015]

Example rows use a made-up client (Northstar Systems). Replace them with your story.

4. Open questions (decide before development starts)

#	Question	Owner	Due	Decision
1	[Does a partial refund count against the 90-day window?]	[Policy owner]	[Date]	[ ]

5. Using these criteria with an AI coding agent

The same rules make criteria an AI agent can build from: one behavior each, real data, an observable result, and the unhappy paths written down. Paste the table above into the agent's task and ask it to write a test for each ID before the code. See also `checklists/agent-ready-requirements.md`.

6. Traceability

Give every criterion an ID and link it to its test case (last column above, and in `templates/traceability-matrix-template.csv`). When someone asks "is this covered?", the answer is one line: AC-04 → TC-014, passed.

ABOUT THE AUTHOR

Prachit Parikh is a business analyst in Apex, NC. He turns what stakeholders ask for into requirements a team can test, checks the data with SQL, and sees changes through UAT. More free templates: github.com/prachitparikh/ba-toolkit. Videos: youtube.com/@PrachitParikh1. Website and portfolio: prachitparikh.github.io.