Test strategy template

Ismail 3000 kit · version 1.0 · copy it and fill it in. Text in italics is guidance to replace.

1. Context

Item	To fill in
Product	*Product name and a one-sentence description*
Users	*Who uses it, how many, on which devices*
Team	*Number of developers, QA, PO; release cadence*
Stakes	*What a bug costs: money, data, reputation, compliance*
Document date and version	

2. Quality goals

Three to five measurable goals, aligned with the stakes.

- No blocking bug on the critical journeys in production.
- A regression caught before release in *X*% of cases.
- Test feedback in under *X* minutes on every pull request.

3. Scope

In scope	Out of scope
Applications, modules, APIs, platforms tested	*What won't be tested, and why*

4. Risk analysis

List the features, rate likelihood and impact from 1 to 3, then multiply. Risks scoring 6 and 9 come first.

Feature	Likelihood (1-3)	Impact (1-3)	Score	Priority
Payment				
Sign-up / login				

5. Critical journeys

The journeys that must never break. They are tested on every release and automated first.

1. *Journey 1*
2. *Journey 2*
3. *Journey 3*

6. Test levels and types

Level	Who	Tools	When
Unit tests	Developers	*Jest, JUnit...*	On every commit
API / integration tests	Developers, QA	*Postman, Karate, Playwright API...*	On every pull request
End-to-end tests	QA	*Playwright, Cypress, Maestro...*	On every pull request (critical journeys), nightly (full suite)
Exploratory testing	QA, PO	Time-boxed sessions	Every sprint
Non-functional	*Who*	*k6, Lighthouse, axe...*	*Frequency*
AI feature testing	QA	*Scenario set, LLM judge*	On every prompt or model change

7. Automation

- **What we automate first**: the critical journeys and regressions we've already hit.
- **What we don't automate**: *very unstable screens, one-off tests...*
- **CI integration**: *CI tool, triggers, target run time*.
- **Stability rule**: a flaky test is fixed or removed from the blocking suite within *X* days.

8. Environments and test data

Environment	Use	Data
Acceptance / staging	*End-to-end and acceptance testing*	*Anonymised data sets, test accounts*

9. Entry and exit criteria

- **A feature enters testing when**: *acceptance criteria are written, the build is deployed, unit tests are green*.
- **A version goes to production when**: *zero open blocking bugs, critical journeys green, acceptance checklist completed*.

10. Defect management

- Tool: *Jira, Linear, GitHub Issues...*
- Ticket template: steps, actual result, expected result, environment, screenshot.
- Severities: **Blocker** (fixed before release), **Major** (fixed within the sprint), **Minor** (prioritised in the backlog).

11. Metrics tracked

Metric	Target	Frequency
Automated critical journey coverage	*X%*	Every sprint
Flaky test rate	*< X%*	Every sprint
Test suite run time in CI	*< X min*	Every sprint
Bugs found in production	*Trending down*	Every month

12. Roles

Role	Responsibilities
QA	*Strategy, automation, acceptance testing, metrics*
Developers	*Unit tests, fixes, test reviews*
PO	*Acceptance criteria, functional sign-off*

This template is free to use in your projects. Find more kits and articles on AI-augmented QA at ismail3000.