

MASFIQUR RAHMAN

QA Work Sample · UAT Tester (Manual / Pre-Production QA)

Prepared for: **UAT Tester — Manual Test Execution & Pre-Production QA**

Focus: Digital journeys · Stories/AC · Test data · Azure DevOps · Pass/fail & daily status

14+ years Manual / Functional / UAT · Web + Mobile · Customer-facing communication

14+	UAT Own	ADO	Web+Mobile
Years QA depth	Cycle ownership	Azure DevOps ready	Cross-platform

About this demo

Illustrative sample artifacts showing how I run **UAT and manual QA** in pre-production and test environments: review designs and acceptance criteria, critically assess customer scenarios, lock test data, execute and document cases in Azure DevOps, track defects, and publish clear daily status with a documented pass/fail recommendation. Anonymized demos — not client-confidential materials. Context: digital customer journey (account → browse → checkout → confirmation) with an upstream payment/order service.

What I bring to this UAT role

- Review solution designs, stories, and AC; understand apps and upstream integrations before execution
- Challenge and strengthen customer test scenarios so coverage is complete and stakeholder-clear
- Proactively create/request test data and confirm it covers every plan scenario
- Manual smoke, functional, and UAT across web and mobile test environments
- Azure DevOps: test cases, cycles, defects with repro + evidence; monitor cycles to exit
- Daily documentation: results, progress, QA status, and quality metrics for customers and delivery

Contents

#	Section	What you'll see
1	UAT operating model	How I own pre-prod cycles end-to-end
2	Story / AC + scenario critique	Gaps I raise on digital journeys
3	Test data + ADO cases	Data matrix and executable cases
4	Defect + daily status	Developer-ready bug and pass/fail report

1. UAT Operating Model (Pre-Production Ownership)

UAT fails when scenarios are vague, data is missing, or pass/fail is opinion. I keep the cycle evidence-based and visible to the customer every day.

Phase	My ownership	Output
Prepare	Review design/docs; map journeys + upstream systems	Risk list; AC gaps; env checklist
Plan	Validate customer scenarios; expand coverage	Agreed test plan in Azure DevOps
Data	Create/request accounts, carts, payments, edge states	Data matrix signed off before start
Execute	Smoke → functional → journey UAT (web + mobile)	Case results + defects with evidence
Decide	Assess UAT criteria against evidence	Pass / Fail / Conditional with rationale
Report	Daily status to customer and delivery	Results, blockers, metrics, next actions

Tools & platforms

Test management: Azure DevOps (Test Plans / Cases / Suites / Bugs)

Also used: Jira · TestRail-style workflows

Evidence: Screenshots · short Loom/video · network notes when useful

Platforms: Web (Chrome/Safari/Edge) · Mobile web / native where in scope

2. Story / AC Review + Customer Scenario Critique

Context (demo): Digital checkout journey — customer completes payment so order is confirmed. Upstream: payment gateway + order service.

Story ID	US-214 · Priority: High · UAT candidate
Persona	Returning customer (logged in) on web and mobile
Story	As a customer, I can complete checkout after payment authorization so my order is confirmed.

Acceptance criteria (as provided)

- **AC1:** Customer can pay with a valid card and see an order confirmation page.
- **AC2:** Order appears in My Orders after successful payment.
- **AC3:** Confirmation email is sent.

Gaps I would raise before UAT starts

Gap	Why it matters	Proposed AC / scenario
Payment timeout / retry	Real networks fail mid-auth	Timeout shows clear message; retry does not double-charge
Partial authorization	Gateway may auth less than cart total	Order blocked; cart/payment state consistent
Upstream order failure	Payment OK but order service 5xx	Refund/void path; no orphan paid-no-order
Mobile confirmation UX	Journey must work on phone	Same AC on mobile viewport / app
Email failure	Customer trust break	In-app confirmation still valid; retry/resend noted

I critically assess customer scenarios the same way: keep what is strong, rewrite what is ambiguous, and add missing negative/integration paths so stakeholders share one clear plan.

3. Test Data Matrix + Azure DevOps Test Cases

I do not start a UAT cycle until data exists for every planned scenario. Below is a sample matrix I would create or request from the customer/environment owners.

Data set	Purpose	Owner / source
Cust-Valid-01	Happy-path checkout (web)	Seeded test account
Cust-Valid-Mob	Same journey on mobile	Same user / second device
Card-OK / Card-Denied	Auth success vs decline	Gateway sandbox
Card-Timeout	Simulate slow/timeout auth	Gateway test token
Cart-Promo-Expired	Promo edge on checkout	Pricing/test data API
Order-Svc-Fault	Upstream 5xx after payment	Chaos flag / mock (if available)
Empty / zero-stock SKU	Block checkout early	Catalog seed

Azure DevOps test cases (excerpt for US-214)

ID	Title	Type	P
TC-01	Smoke: login → cart → checkout page loads (web)	Smoke	P0
TC-02	Happy path: valid card → confirmation + My Orders	Functional / UAT	P0
TC-03	Happy path on mobile viewport / device	Functional / UAT	P0
TC-04	Declined card: clear error; cart retained; no order	Negative	P0
TC-05	Payment timeout: message + safe retry (no double charge)	Negative	P0
TC-06	Upstream order failure after auth: no orphan payment	Integration	P0
TC-07	Confirmation email received (or in-app + resend noted)	Functional	P1
TC-08	Regression: promo + tax totals match summary	Regression	P1

In Azure DevOps: cases linked to US-214, assigned to the UAT suite/cycle, executed with Pass/Fail/Blocked, and defects linked back to failed steps.

4. Sample Defect + Daily UAT Status (Pass/Fail)

BUG — Payment succeeds but order missing after upstream timeout

Severity	Critical · Azure DevOps Bug linked to US-214 / TC-06
Environment	UAT / Pre-Prod · Chrome + iOS Safari · Build 3.4.2-rc.2
Evidence	Screenshots (paid UI vs empty My Orders) · 45s Loom · payment ref + network 504

Steps

- Login as Cust-Valid-01; add in-stock item; proceed to checkout.
- Pay with Card-OK while Order service is forced to timeout / 504 (test flag).
- Observe checkout result UI; open My Orders; check confirmation email.

Expected

Clear failure message; payment voided/refunded or pending with tracking; no confirmation of a completed order; no charge without order record.

Actual

UI shows "Payment successful" briefly, then generic error. My Orders empty. Payment reference remains Authorized in gateway sandbox. Customer cannot tell if order exists.

Impact on UAT exit

Blocks UAT pass for US-214 — digital journey cannot be signed off until payment/order consistency is fixed and retested on web and mobile.

Daily UAT status sample (customer-ready)

Metric	Value
Cycle	UAT Sprint-12 · Checkout journey (US-214)
Executed today	12 cases (8 Pass / 3 Fail / 1 Blocked)
Cumulative	28 / 40 planned (70%)
Open defects	2 Critical · 1 High · 2 Medium
Blockers	Order-svc fault path; mobile confirmation email delay
UAT recommendation	FAIL (conditional) — P0 journey not exit-ready until Critical defects verified
Tomorrow	Retest Critical fix; complete mobile suite; refresh data for promo edges

Masfiqur Rahman · Full-Stack QA Engineer · Manual / Functional / UAT

14+ years · Azure DevOps · Web + Mobile · Customer-facing status & pass/fail ownership

Portfolio: <https://www.masfiqur.com> · Resume: <https://www.masfiqur.com/cv/MD-Masfiqur-Rahman-CV.pdf>

Happy to adapt these templates to your Azure DevOps project, environments, and UAT exit criteria.