

1. Purpose & Quality Philosophy

هدف این سند پاسخ به یک سؤال ساده است:

«چطور مطمئن می‌شوید هر feature جدید، سیستم را خراب نمی‌کند؟»

فرض ما:

بدون تست → هر deploy یک قمار

با تست → رشد قابل پیش‌بینی

2. Quality Ownership Model

کیفیت «وظیفه QA تنها» نیست.

Role	Responsibility
Engineer	Unit + Integration tests
QA	E2E, regression, release sign-off
Product	Acceptance criteria
DevOps	Test automation infra

اگر کسی گفت QA «بعداً تست می‌کند» → این تیم fail است.

3. Test Pyramid Strategy

E2E کم، حیاتی

Integration متوسط

Unit Tests زیاد، سریع

هدف:

- بیشترین پوشش با کمترین هزینه
- کمترین تست شکننده

4. Unit Testing Strategy

Scope

- Business logic
- Ranking algorithms
- Pricing rules

Edge cases •

Tools

Backend: Jest / PyTest •

Frontend: React Testing Library •

Coverage Target

Core logic: $\geq 80\%$ •

Overall codebase: $\geq 65\%$ •

Coverage عدد نیست؛ سیگنال ریسک است.

5. Integration Testing

What We Test

API \leftrightarrow Database •

Auth flows •

Payment & billing logic •

External service mocks •

Why

Unit کافی نیست •

بیشترین باگها اینجا رخ می‌دهند •

6. End-to-End (E2E) Testing

Critical User Journeys

Signup \rightarrow onboarding •

Create campaign •

Ranking execution •

Payment confirmation •

Tools

Playwright / Cypress •

Policy

فقط مسیرهای حیاتی •

شبانه اجرا می‌شوند •

block release اگر fail شوند •

7. Regression Testing Strategy

Trigger

- Every release
- High-risk refactor
- Infra changes

Coverage

- Core flows
- Known fragile areas

Regression = بیمه رشد

8. Security Testing

Automated

- Dependency vulnerability scan
- Static code analysis
- Secrets detection

Manual

- Auth edge cases
- Role escalation attempts
- Input sanitization

Security با feature فرق ندارد؛ باید تست شود.

9. Release Process & Quality Gates

Release Flow

1. Code freeze
2. Full test suite
3. QA sign-off
4. Canary deploy
5. Production release

Quality Gates

Gate Must Pass	
Unit tests	✓
Integration tests	✓
E2E critical	✓
Security scan	✓

10. Test Evidence (What We Show)

Artifacts Provided	
CI test reports	•
Coverage dashboards	•
E2E execution logs	•
Failed test history	•
ما «می‌گوییم تست داریم» نه؛ نشان می‌دهیم.	

11. Bug Management & Triage Policy

Bug Classification	
Severity	SLA
Critical	<24h
High	<72h
Medium	Next sprint
Low	Backlog
Root Cause Analysis	
Why test missed it?	•
Which layer failed?	•
Prevention action	•

12. Flaky Test Policy

Flaky test = production risk	•
Immediate quarantine	•

- Fix or delete

تست بد، بدتر از نبود تست است.

13. Quality Metrics (Tracked Monthly)

Metric	Target
Escape defects	<5%
Test pass rate	>95%
Rollback rate	<5%
Mean bug fix time	<48h

14. Scaling Quality with Growth

As product grows:

- Test suites parallelized
- Contract testing added
- Synthetic monitoring
- Load tests before scale milestones

15. Investor / CTO Takeaway

این تیم کیفیت را با فرایند و داده کنترل می‌کند، نه با امید.

نبود تست = هزینه رشد نمایی
وجود تست = رشد خطی و قابل پیش‌بینی