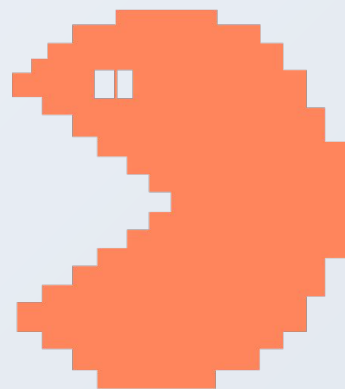




The Startup QA Readiness

Checklist



Your 30-Day Implementation Roadmap from Chaos to Quality

Transform your testing in 30 days



hope it works



know it works

Do 30 days sound too harsh? Make it 90.
Still, we don't advise planning for more
than 90 days — keep it focused.

**Need help turning this
checklist into action?**

TestFort specializes in QA for scaling
startups.

Let's talk!

[\[email\]](#)

[Free QA Audit: \[website\]](#)

[Book a consultation: \[calendar link\]](#)

Week 1:



The reality audit

*Understand where you really stand
(spoiler: it's probably worse than you think)*

Days 1-3: Current State Assessment

Time to face the music. You know things aren't perfect, but you need real numbers to make a case for QA. So, let's create a baseline.

Warning: These numbers might hurt, but they're your ammunition for getting buy-in and budget.

Testing Reality Check

- ☐ Document who currently tests each feature/component
- ☐ Calculate average bugs reaching production per week: _____
- ☐ Track hours spent on hotfixes this week: _____
- ☐ Count customer-reported bugs vs. internally found: ____ / ____
- ☐ List the last 5 critical bugs that reached production

Time Analysis

- ☐ Developer time on bug fixes vs. features: _____% vs _____%
- ☐ Average time from bug report to fix: _____ hours
- ☐ Deployment delays due to quality concerns: _____ times/month
- ☐ Support tickets related to bugs: _____ / week

Cost Calculator

- ☐ Lost revenue from bugs this month: \$ _____
- ☐ Developer hours on debugging: _____ × hourly rate = \$ _____
- ☐ Customer churn attributed to quality issues: _____%
- ☐ Delayed features due to bug fixing: _____

Red Flag Checklist (Check all that apply)

- ☐ "It works on my machine" is said daily
- ☐ Customers find bugs before you do
- ☐ Same bugs keep reappearing
- ☐ No one wants to deploy on Fridays
- ☐ Production is your testing environment
- ☐ "Quick fixes" take days ☐ You've lost a customer due to bugs

Score

_____ / 7 red flags

 4+ means immediate action needed

Week 1. Days 4–7: Critical Path Mapping

Now let's find your weak spots. Not all bugs are created equal — a typo in your footer is annoying, but a bug in your payment flow is catastrophic. These few days are about identifying what absolutely cannot break. Think of it as creating a map of your product's vital organs. You'll test everything eventually, but these paths get bodyguards first.

Revenue-Critical Paths (List and rate importance 1–5)

- ☐ User Registration → First Value: _____
- ☐ Free Trial → Paid Conversion: _____
- ☐ Payment Processing Flow: _____
- ☐ Core Feature Usage: _____
- ☐ Data Import/Export: _____

User Journey Mapping

Map top 5 user journeys with steps:

1. Journey: _____
Steps: _____ → _____ → _____ → _____
2. Journey: _____
Steps: _____ → _____ → _____ → _____
3. Journey: _____
Steps: _____ → _____ → _____ → _____
4. Journey: _____
Steps: _____ → _____ → _____ → _____
5. Journey: _____
Steps: _____ → _____ → _____ → _____

Integration Points Inventory



- ☐ Payment processors: _____
- ☐ Email services: _____
- ☐ Analytics tools: _____
- ☐ Third-party APIs: _____
- ☐ Database connections: _____
- ☐ Authentication services: _____

Risk Assessment Matrix

Component	Impact if Fails (1-5)	Failure Frequency	Priority
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Need help turning this checklist into action?

TestFort specializes in QA for scaling startups.

Let's talk!

[email] Free QA Audit: [website] Book a consultation: [calendar link]

Share your progress: #StartupQA30Days

Week 2:



Stop the bleeding

Quick wins that prevent disasters

Days 8–10: Smoke Test Implementation

Remember when pilots do their walk-around before takeoff? That's what we're building here — a quick sanity check that takes 15 minutes but saves you from 3 AM emergency calls. It's not fancy, it might even be a Google Doc at first, but it's the difference between



we think it works



we checked, it works

Basic Smoke Test Checklist (Customize for your product)

- ☐ Homepage loads in <3 seconds
- ☐ Login functionality works
- ☐ Password reset email sends
- ☐ Main navigation accessible
- ☐ Search returns results
- ☐ Core feature #1 functions: _____
- ☐ Core feature #2 functions: _____
- ☐ Core feature #3 functions: _____
- ☐ Payment page loads
- ☐ Checkout process completes
- ☐ API health endpoint responds
- ☐ Database connection active
- ☐ Mobile view renders correctly
- ☐ No JavaScript console errors
- ☐ Analytics tracking fires

Smoke Test Setup

- ☐ Create written checklist in shared doc
- ☐ Assign primary tester: _____
- ☐ Assign backup tester: _____
- ☐ Set execution trigger:
- ☐ Pre-deploy
- ☐ Post-deploy
- ☐ Both
- ☐ Time the full test run: _____ minutes (target: <15)
- ☐ Create Slack channel for results: # _____
- ☐ Document how to run each test step

Quick Win Automations

- ☐ Set up uptime monitoring (UptimeRobot/Pingdom)
- ☐ Create basic API health check
- ☐ Implement deployment notifications
- ☐ Add rollback procedure documentation



Week 2. Days 11-14: Error Tracking Setup

Right now, you probably find out about bugs when angry customers email you. That changes today. Error tracking is like having security cameras in your code — you'll see problems as they happen, not days later.

Tool Selection & Setup

- ☐ Choose error tracking tool:
- ☐ Sentry (Recommended for most)
- ☐ Rollbar
- ☐ Bugsnag
- ☐ LogRocket
- ☐ Other: _____

Configuration Checklist

- ☐ Install error tracking SDK
- ☐ Set up environments (dev/staging/prod)
- ☐ Configure user context capture
- ☐ Set up source maps
- ☐ Enable release tracking
- ☐ Create error grouping rules
- ☐ Set sampling rate: _____%

Alert Configuration



- ☐ Connect to Slack: # _____
- ☐ Set up email alerts: _____
- ☐ Configure alert rules:
 - ☐ New error types
 - ☐ Error spike detection
 - ☐ High-priority errors
 - ☐ Payment-related errors
 - ☐ Create on-call rotation schedule
 - ☐ Document escalation process

Baseline Metrics (Record current state)

- ☐ Daily error rate: _____
- ☐ Unique errors per week: _____
- ☐ Most frequent error: _____
- ☐ Average error resolution time: _____
- ☐ User-affecting errors: _____%

Need help turning this checklist into action?

TestFort specializes in QA for scaling startups.

Let's talk!

[email]

Free QA Audit: [website]

Book a consultation: [calendar link]

Share your progress: #StartupQA30Days

Week 3:



Build the foundation

From reactive to proactive

Days 15–17: First Automated Test

Manual testing is like washing dishes by hand – necessary, but inefficient at scale. We're going to automate just **one critical test** this week. Not ten, not the entire app – just one that runs perfectly every time. This single test will run hundreds of times and catch issues while you sleep. Start small, but start.

Test Framework Setup

- ☐ Choose testing framework:
- ☐ Cypress (Recommended for web)
- ☐ Playwright
- ☐ Selenium
- ☐ Puppeteer
- ☐ Other: _____

Environment Preparation

- ☐ Set up test environment
- ☐ Create test data setup/teardown
- ☐ Configure CI integration
- ☐ Set up test reporting
- ☐ Document local setup steps

First Test Implementation

- ☐ Select most critical path: _____
- ☐ Write test for happy path
- ☐ Add negative test cases
- ☐ Include edge cases
- ☐ Test cross-browser (minimum 2)
- ☐ Add mobile viewport tests
- ☐ Set up screenshot on failure
- ☐ Configure retry logic

Success Criteria

- ☐ Test runs in <5 minutes
- ☐ Covers 80% of critical path
- ☐ Zero false positives in 5 runs
- ☐ Clear failure messages
- ☐ Automated execution on push

Days 18–21: QA Workflow Design

Creating your quality assembly line. Right now, your testing is probably ad-hoc – different every time, depending on who's doing it. We're going to create a repeatable process that everyone follows. Think of it as your team's quality recipe – same ingredients, same steps, consistently good results. No more "I thought someone else tested that."

Process Definition Define "Definition of Done":

- ☐ Code reviewed
- ☐ Unit tests pass
- ☐ Integration tests pass
- ☐ QA approved
- ☐ Documentation updated
- ☐ _____

Testing Stages

- ☐ Development testing requirements: _____
- ☐ Staging testing requirements: _____
- ☐ Production testing requirements: _____
- ☐ Rollback criteria: _____

Responsibility Matrix



Stage	Who	When	Blocking?
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Bug Management Process

- ☐ Bug report template created
- ☐ Severity levels defined (P0-P3)
- ☐ SLA per severity set:
P0: _____ hours P1: _____ hours P2: _____ days P3: _____ days
- ☐ Bug triage meeting scheduled: _____
- ☐ Bug tracking system configured

Need help turning this checklist into action?

TestFort specializes in QA for scaling startups.

Let's talk!

[email] Free QA Audit: [website] Book a consultation: [calendar link]

Share your progress: #StartupQA30Days

Week 4:



Measure & scale

Data-driven decisions for long-term success

Days 22–24: Metrics & KPIs

Numbers don't lie (even when they're painful). After three weeks of changes, it's time to see if this is actually working. Some metrics will improve immediately, others might get worse before they get better (you're finding more bugs – that's good!). These numbers become your QA report card and your justification for further investment.

Quality Metrics Tracking

- ☐ Bugs caught pre-production: ____/week
- ☐ Bugs escaped to production: ____/week
- ☐ Escape rate: ____%
- ☐ Average bug fix time: ____ hours
- ☐ Test coverage: ____%
- ☐ Test execution time: ____ minutes
- ☐ Failed deployments: ____/month

Team Health Metrics

- ☐ Developer satisfaction survey (1-10): ____
- ☐ Deployment confidence (1-10): ____
- ☐ Time spent firefighting: ____%
- ☐ Feature velocity change: +/- ____%



Customer Impact Metrics

- ☐ Customer-reported bugs: _____ /week
- ☐ Support tickets (bug-related): _____ /week
- ☐ Bug-related churn: _____ %
- ☐ NPS impact from bugs: _____

ROI Calculation

- ☐ Hours saved on debugging: _____
- ☐ Prevented outages: _____
- ☐ Reduced customer churn: %
- ☐ Value delivered: \$

Days 25–28: Hiring/Outsourcing Decision

Time for the big decision. You've got three weeks of data, you know what's working and what's not. Now you need to decide: build an in-house team or bring in external help? There's no universal right answer, but after these exercises, you'll know what's right for YOUR startup. This decision will shape your QA strategy for the next year.

Needs Assessment

- ☐ Required QA hours/week: _____
- ☐ Specialized skills needed: _____
- ☐ Budget available: \$_____ /month
- ☐ Timeline urgency (1–10): _____

In-House Hire Evaluation

- ☐ Can afford \$70-90K/year? Yes/No
- ☐ Have 2-3 months to hire? Yes/No
- ☐ Need daily collaboration? Yes/No
- ☐ Have mentorship capacity? Yes/No
- ☐ Score: ____ /4 (3+ = consider in-house)

Outsourcing Evaluation

- ☐ Need expertise immediately? Yes/No
- ☐ Variable testing needs? Yes/No
- ☐ Budget \$2-5K/month? Yes/No
- ☐ Comfortable with remote? Yes/No
- ☐ Score: ____ /4 (3+ = consider outsourcing)

Vendor/Candidate Evaluation

- ☐ Technical assessment completed
- ☐ Communication skills verified
- ☐ Domain knowledge checked
- ☐ References contacted
- ☐ Trial period defined: ____ days
- ☐ Success criteria set

Days 29–30: Quarter 2 Planning

Victory lap and battle plan. You made it! Take a moment to appreciate how far you've come in 30 days. But this is just the beginning. Based on everything you've learned, let's plan the next 90 days. What worked? What didn't? What nearly gave you a heart attack? Your second quarter of QA will be 10x more effective because you're not guessing anymore.

Retrospective Questions

- ☐ Biggest win this month: _____
- ☐ Biggest challenge: _____
- ☐ Most surprising finding: _____
- ☐ Process that needs refinement: _____

Next Quarter Priorities (Choose top 3)

- ☐ Expand test automation coverage to _____%
- ☐ Reduce bug escape rate to _____%
- ☐ Implement performance testing
- ☐ Add security testing
- ☐ Create regression test suite
- ☐ Build API test automation
- ☐ Set up load testing
- ☐ Implement visual regression testing
- ☐ Other: _____

Resource Planning

- ☐ Additional tools needed: _____
- ☐ Training required: _____
- ☐ Budget request: \$ _____
- ☐ Headcount needs: _____

Success Metrics for Q2

- ☐ Target bug escape rate: _____%
- ☐ Target test coverage: _____%
- ☐ Target deployment frequency: _____/week
- ☐ Target MTTR: _____ hours

Need help turning this checklist into action?

TestFort specializes in QA for scaling startups.

Let's talk!

[email] Free QA Audit: [website] Book a consultation: [calendar link]

Share your progress: #StartupQA30Days

Bonus:



Quick reference section

Emergency QA Toolkit (Free/Cheap)

You're not trying to build NASA-level testing in 30 days. You're building a safety net that catches the bugs that would hurt your business most. Every test you add, every bug you catch before production, is a small victory. Stack enough small victories, and you've got a quality product.

Must-Have Tools

BrowserStack — Free tier: 100 min/month

Postman — Free for 3 users

GitHub Actions — 2,000 minutes/month

Sentry — 5K events/month free

Cypress — Open source

Worth Paying For

TestRail — \$30/user/month

Percy — \$149/month

Datadog — From \$15/host

Red Alert Triggers (Implement QA Now)

- ✓ More than 5 customer-reported bugs/week
- ✓ Any security or payment bug in production
- ✓ Same bug appearing 3+ times
- ✓ Deployment rollback in last month
- ✓ Developer spending 30%+ time on bugs

QA Implementation Success Formula

Week 1: Measure everything (it's worse than you think)

Week 2: Stop critical bleeding (smoke tests save lives)

Week 3: Automate one thing well (perfect is the enemy of done)

Week 4: Decide and commit (data doesn't lie)

Ask yourself every morning: "If we had 10x the users today, what would break first?"

That's where you focus your QA efforts. The goal isn't to test everything — it's to test what matters most.



Time for your commitment

I _____, commit to implementing QA because:

- ☐ Our customers deserve better
- ☐ Our developers deserve better
- ☐ Our business depends on it
- ☐ Quality is not optional at scale

Start Date: _____

First Milestone: _____

Accountability Partner: _____

Share your progress: #StartupQA30Days

TestFort: Your Partner for Innovative QA Solutions



CMMI DEV3 Maturity Level

ISTQB Certified Testers

ISO 27001:2013 Security Standard

23+

years of market expertise

Product mindset:

result-oriented approach

Proven track record

with industry leaders

80+

automation projects complete

Tailored solutions

for short and long-term projects

Deep understanding

of industry-specific challenges

Partner with TestFort for QA excellence

Let's ensure your software quality
matches your innovation.

Contact Us

Phone: +1 310 388 93 34

USA, UK, Ukraine

<https://tesfort.com>

