SINCE 1994

Shoestring *manual testing.*

WHEN YOU CAN'T AFFORD AUTOMATION, YET · REX BLACK, INC.

Total automation is a *mirage*.

For most real projects, at least one obstacle kills the dream of "start the test, leave it running, come back for results."

The answer is not to give up on testing.
The answer is a **well-run manual program.**

WHAT THIS IS ABOUT

In plain English.

You don't have the budget, tools, headcount, or stable system under test that automation needs. The product ships anyway. How do you run *real* testing on a shoestring?

This talk answers:

- **How to size the manual effort honestly** — with two rules of thumb your CFO will believe.
- **Where to find good testers cheaply** — and what each hiring pool's weakness is.
- **How to train, shift, and supervise** a lean manual team.

MANUAL VS. AUTOMATED

The field test.

AUTOMATED

Tester **starts the run**, walks away, comes back for results.
Unattended execution. Machine compares actual to expected.

MANUAL

Tester enters data, observes behaviors, **actively runs the tests.**
Tools are fine. Tools ≠ automation.
Unattended is automation.

Five obstacles.

- **SUT changes frequently** — scripts break faster than you write them.
- **No budget or calendar** for script development.
- **No suitable tool** at a justifiable price.
- **Team lacks automation skill** or experience.
- **Critical risks aren't amenable.** Usability, localization, error handling, installation, docs — **cheaper and more honest manually.**

SIZING

Two rules of thumb.

Nobody wants to hear them.

RULE 1

Six productive test hours per 8–10 hour day.

Testers do **not** test for eight hours.
Testers test for **six**, and spend the other two fighting their environment.

Bug reporting, status documentation, communication, breaks, blocking issues, debugging — it's part of the job, **not an excuse.**

RULE 2

25% downtime on good builds.
75% downtime on bad ones.

Four days of calendar to get one day of test coverage, on a bad build.

Budget the worst case up front. **Do not promise a schedule without it.**

One manager.
A few engineers.
Many technicians.

TEAM STRUCTURE

Three roles. *Leveraged.*

TEST MANAGER

Overall responsibility for the team, the schedule, and the release recommendation. **One per program.**

TEST ENGINEERS

Technical leadership. Design tests. Troubleshoot hardest failures. **Supervise several technicians each.**

TEST TECHNICIANS

Most of the actual execution. **Where the coverage comes from.**

THE LEVERAGE

One or two engineers ramp and supervise several technicians — coverage expands without **tripling salary cost.**

Career ladder: technician → engineer.

HIRING

Four cheap sources.

STUDENTS (CS / ENGINEERING)

Eager. Cheap.
Sometimes unavailable.

MOONLIGHTERS

Real experience from other jobs.
Sometimes distracted.

SUPPORT CROSSOVERS

Understand customer problems.
Sometimes too solution-focused.

DATA-ENTRY STAFF

Fluent computer users.
Sometimes low-curiosity.

Pick the mix that fits your product and your office culture.

TRAINING

Universal skills *vs.* company rituals.

UNIVERSAL — TRAIN HARD

Manual test-case execution.
Bug reporting.

Written docs, training, coaching.
Every technician. Every source.

COMPANY-SPECIFIC — DOCUMENT

Network login, badge, email, phone system, test repo.

Documentation or a mentor.
Not a skill. A ritual.

The technician who **cannot reproduce and document a bug** is broken in a way a second badge tour does not fix.

THE SINGLE MOST IMPORTANT SKILL

Ten-step bug report.

Every technician has to be **fluent** in one thing above all others.

Investigation half: structure · reproduce · isolate · generalize · compare.
Writing half: summarize · condense · disambiguate · neutralize · review.

Grammar and spelling count. A poorly written report can render a technically correct one *ridiculous*.

See */resources/qa-library/bug-reporting-process* for the full checklist.

MULTIPLE SHIFTS

Same salary envelope. *More coverage.*

Night (~4 PM–1 AM) and graveyard (~midnight–9 AM) shifts stretch a shoestring team.

WHY

- **Schedule** — more tests per day.
- **Resources** — one hardware platform covers two or three eight-hour blocks.
- **Preference** — some technicians

GET IT RIGHT

- **Explicit at interview.**
- **Undesirable-hours bonus.**
- **Overlap shifts for hand-offs.**
- **Invest in team atmosphere** — out of

WHAT NOT TO RUN MANUALLY

The line you do not cross.

MANUAL WORKS

Functional · use cases · error handling · localization · UI · configs · operations · date/time · installation · documentation.

Humans judge better than machines here.

AUTOMATION-ONLY

Monkey/random · load/volume/capacity · reliability/MTBF · code coverage · performance · standards compliance.

Running these manually either doesn't work or *misleads* people about coverage.

Even a shoestring program buys or builds **some** automation for the right column.

SELL IT TO MANAGEMENT

It's a *solution*, not a concession.

Bring a **plan**, not a request for more money.

- **Flexibility and responsiveness** — technicians ramp fast.
- **Cost and coverage** — cheaper bodies = more coverage on the same budget.
- **Schedule** — test cycle time goes down.
- **Staff retention** — test engineers burn out less when they have technicians to delegate to.
- **Career development** — helps inexperienced people jump-start technical careers.

Enough signal to ship safely.

That is the goal.

TAKEAWAYS

Four to hold against.

- **Total automation is a goal, not a precondition.** Manual program bridges "nothing" and "right automation, right time."
- **Budget the overhead before the test count.** 6 hours, 75% downtime. Plan them in.
- **Leverage from the org shape.** Manager + engineers + many technicians. Flat senior teams run out of money before tests.
- **Pick manual for what humans judge best.** Usability stays manual. Load stays automated. Even on a shoestring.

> *Using manual testing inappropriately can mislead people about the extent of test*

SINCE 1994

Thank you.

REX BLACK, INC. · REXBLACK.COM/RESOURCES/TALKS/SHOESTRING-MANUAL-TESTING