SINCE 1994

Charting the progress of *system development.*

FIVE BUG-DATA CHARTS EVERY TEST LEAD SHOULD RUN EVERY WEEK · REX BLACK, INC.

Bug data is the most *honest* dashboard you have.

Status reports are green because the author wants them to be.
Bug data is what it is.

Five charts. One hour a week.
That is the whole method.

WHAT THIS IS ABOUT

In plain English.

If you run or *oversee* a software project, your bug tracker already has the most honest status report you'll ever get. You just have to plot it.

This talk shows five simple charts that answer four questions:

- **Is the product stabilizing?** — or still getting worse every week.
- **Is the dev team keeping up?** — or quietly falling behind.
- **Where are the mistakes being made?** — requirements, code, environment.
- **Which part of the product is in the most trouble?** — where to spend next.

WHY CHART BUG DATA

Two reasons.

ASSESS

Product stability. Defect-removal trend. Root-cause distribution. Hot-spot subsystems.

COMMUNICATE

Summarize facts and trends — not incident counts. One chart, one argument. Management can act.

Manage the key indicators, not the crisis du jour.

THE WORKED EXAMPLE

SpeedyWriter.

Three cycles each of **component**, **integration**, and **system** test. Roughly **105 bugs** over two months. First customer ship: **13 September**.

COMPONENT TEST

7/19 — 25 bugs
7/26 — 20 bugs
8/2 — **5 bugs**

INTEGRATION TEST

8/2 — 20 bugs
8/9 — 15 bugs
8/16 — **5 bugs**

System test: 8/16 — 10 · 8/23 — 5 · 8/30 — **0**. The zero is the first real stability signal.

Two columns. *Opened date.* *Closed date.*

Everything in this deck comes from those two columns plus a root-cause tag and a subsystem tag at close.

Bug reports approximate underlying bugs linearly — roughly **27% high** because of duplicates and rejections. The charts hold up.

CHART 1 · THE FLAGSHIP

Cumulative opened vs. *cumulative closed.*

Run `COUNTIF` by date. Two running totals. Plot both on the same axis.

CHART 1 · READING IT

The gap *is the quality backlog.*

OPENED CURVE

Flattens as the system stabilizes and the team finds what it can find.

CLOSED CURVE

Converges to the opened curve as the product becomes customer-ready.

Derivative of the opened curve is your **daily find rate** — also worth plotting. Milestones in the project show up as changes in shape.

CHART 1 · TROUBLE PATTERN 1 OF 3

Endless bug discovery.

Shape. The opened curve never flattens. Through the planned release date, test is still finding bugs at near-peak rate.

Reading. The product is not stabilizing. Either the test team is exploring net-new territory each cycle, or every build is introducing new defects faster than the old ones are fixed.

Remedy. Stop the feature stream. Hold an integration line. Run a full regression against the last good build. Compare.

Ignored bug reports.

Shape. The opened curve flattens normally. The closed curve lags farther and farther behind. The gap widens instead of closing.

Reading. The development team is not keeping up — understaffed, overcommitted to features, or not incentivized to close.

Remedy. Rebalance dev effort toward close-out, or formally accept the quality backlog as release risk. *Do not hide it.*

Poor report management.

Shape. Both curves are noisy. The closed curve jumps around. Bugs keep re-opening.

Reading. The bug-report process itself is breaking down — sloppy verification, disputed closes, missing regression tests.

Remedy. Tighten close-out criteria. Add a re-verify step before close. Run the *bug-reporting-process* checklist with the team.

CHART 2 · DEVELOPER RESPONSIVENESS

Closure period.

The **age of a bug** at the moment it closes.
Daily and rolling.

Four mechanical steps.

1. For each closed bug: `closure_period = closed_date − opened_date`.
2. **Sum** closure periods by closed_date.
3. **Count** the bugs closed on each date.
4. **Daily** = sum ÷ count. **Rolling** = running sum ÷ running count.

All four are pivot-table primitives in any modern spreadsheet.

CHART 2 · READING IT

A stable closure period is a *smooth* rolling line.

Daily values bounded at the top by **two to three test-cycle durations.**

On SpeedyWriter, each cycle is a week — so a well-run project sees daily closure periods mostly under three weeks.

Exploding closure period means dev is not keeping up, regardless of how green the status report is.

CHART 3 · WHERE THE MISTAKES ARE

Root cause breakdown.

Tag every bug at close.
Requirements · Design · Coding · Configuration · Environment · Documentation · Test case.

CHART 3 · WHAT IT DRIVES

Course corrections now. Process improvements next.

DURING THE PROJECT

A spike in requirements-driven bugs says the spec process needs a round. A spike in coding-driven bugs says code review needs a round.

AFTER THE PROJECT

The distribution is the input to the next cycle's development-process improvement plan. Phases, not names.

CHART 4 · HOT SPOTS

Subsystem breakdown.

Where there is one bug,
there is often another.

CHART 4 · TWO ACTIONABLE PATTERNS

Two things fall out.

MORE TESTING HERE

Subsystems with the most bug reports need more test investment. The Pareto holds on almost every system you'll ever ship.

MORE ENGINEERING HERE

Error-prone subsystems are candidates for development-process improvement too. Preventing upstream is cheaper than finding downstream.

Five charts.

One hour a week.

The shape carries the signal.

- **Totals lie. Curves do not.**
- Learn the three trouble patterns — you'll spot them for the rest of your career.
- Track the **quality backlog** (opened − closed), not just the find rate. It trends to zero before release — or nothing else matters.

Tag at close. Every bug.

- **Closure period** tells you if developers are keeping up.
- **Root cause** at close tells you where the mistakes are being made — and what to fix for next release.
- **Subsystem** at close tells you where to re-invest test and engineering effort now.

Anything more is a project for the data team.
Anything less is inadequate.

SINCE 1994

Thank you.

REX BLACK, INC. · REXBLACK.COM/RESOURCES/TALKS/CHARTING-DEFECT-DATA