



Manual to Automated: What the Change Is Worth

Value Calculators · <https://docs.functionize.com/hc/en-us/articles/43795417732119-Manual-to-Automated-What-the-Change-Is-Worth>

For a team whose regression testing is mostly done by hand. This calculator values the hours you stop spending and the defects you stop shipping, then sets both against what the change costs you in year one — including the effort to build the suite, which most models leave out.



HOW THE MODEL IS BUILT

1

Your baseline — six numbers, none of them ours

Releases per year, regression cycles per release, manual hours per cycle, your blended hourly cost, escaped defects last quarter, and how many critical journeys are covered today.

2

Three assumptions that decide everything

How much of the manual suite is genuinely automatable, what that does to escaped defects, and what it costs to build. **These are the three a reviewer will challenge**, so the defaults are deliberately conservative.

3

The case, calculated

Hours displaced and defect cost avoided, set against year-one build and licence cost.

WHAT THE DEFAULTS PRODUCE

27%

year-one return on investment

5.9

months to payback

Those defaults are deliberately unflattering — 60% of the suite automatable, a 25% reduction in escaped defects. A business case that survives pessimistic assumptions survives a finance review. Put your own numbers in and it will usually look better.



Six numbers of yours, three assumptions, one business case.

Fifteen inputs. About forty minutes with someone from the delivery team.

Download the calculator

Free to download, edit and put your own name on. Every assumption is a cell you can change — nothing is hidden in a formula. **Prefer Google Sheets?** Download the workbook, then in Sheets choose *File* → *Import* → *Upload*. Every formula is written to survive the import.

Excel workbook (.xlsx) · 69 KB

How to use it (PDF) · 117 KB

What it asks you for

Your baseline	Releases per year, regression cycles per release, manual hours per cycle, your blended hourly cost, escaped defects last quarter, and how many critical journeys are covered today. None of it comes from Studio.
Three assumptions	How much of the manual suite is genuinely automatable, what that does to escaped defects, and what it costs to build. These three decide the answer, and each carries a defensible range.
The cost	Your annual Functionize cost, and a discount rate for the three-year view.

What it gives you back

Gross annual benefit, year-one net benefit, return on investment, payback period in months, and a three-year net present value so the number can sit beside other investments on the same slide.

The sheet to take into the review

Sensitivity. It shows the case at three levels of optimism, the case with *no defect reduction at all*, and the automatable share at which year one stops paying for itself. Being asked "what if you are wrong?" and having the answer already on the page is worth more than a bigger headline number.

The three arguments you will have

- **"You will not really free those hours."** Often fair. Testers usually get redeployed rather than removed. Present the hours as capacity released and say what it gets spent on.
- **"The defect reduction is guesswork."** Partly, which is why the sensitivity sheet shows the case assuming none. If that line is still positive, the defect argument is upside rather than the foundation.
- **"Maintenance will eat it."** The answer is to measure it rather than argue. Maintenance share is one of the five KPIs in [What to Measure: QE Metrics and KPIs with Studio](#), and it is the number to report monthly.

Where the inputs come from

Release cadence and regression hours: your release process, and the team — ask them, their figure beats an estimate. Escaped defects: your incident or ticket system, over a full quarter rather than a good month. Cost of a defect: engineering time plus support plus rework; your finance team may already have a figure.

Deciding what is genuinely automatable is its own question, and [What Makes a Good Automation Candidate](#) has the six-question scoring behind it.

What the platform costs you

You do not type in a license figure. The **Platform and AI cost** sheet works it out from your plan tier and what you expect to consume, and the business case reads the answer from there.

Three things it asks for. Your **plan** — monthly fee, credits included, and the price of an additional credit block, all as input cells so they can be checked against current pricing or replaced with the figures on an Enterprise order form. Your **consumption** — tests built per month, runs per month, and repairs per month, priced at the published illustrative rates of 200,000 credits for a new mid-length test and 4,000 for a run. And, optionally, your **LLM spend**, if you drive Studio from Claude Code, Cursor or another tool over MCP.

Two things worth knowing before you present it. The MCP block is **additional, not a substitute** — driving Studio from your own AI tool does not reduce credit consumption, because the same agent work and the same test runs still happen. And **month one is not the steady state**: building a suite from nothing costs far more credits than running it afterwards, so the first invoice will not look like the twelfth. Say that to whoever approves the spend before they see it, not after.

If the annual figure comes out small next to the labor saving, that is the published pricing working as intended rather than an error — but it does mean the case now rests almost entirely on your own effort and defect numbers. Take the sensitivity sheet into the review, not this one. There is an override cell for Enterprise customers with a negotiated annual figure.

See [How Credits Work](#) for where the credit rates come from.

After the decision

Put a date in the diary for one quarter out and compare four numbers against the baseline: escaped defects, critical journey coverage, maintenance share, and manual regression hours per release. If those four are moving, the model was right enough.

Already automating, and comparing tooling rather than replacing manual effort? Use [Baseline and Improve: Valuing a Change in Tooling](#) instead.

Related

- [Value Calculators: Building the Business Case](#) — the other calculator, and how both are built
- [What to Measure: QE Metrics and KPIs with Studio](#) — where each input comes from

- [What Makes a Good Automation Candidate](#) — the scoring behind the automatable share
- [Choosing What to Automate First](#) — what to build once the case is approved