How to Use the Manual to Automated Calculator

A one-sitting guide to building the business case

This calculator answers one question: what is it worth to stop testing your regression suite by hand? It values the hours you stop spending and the defects you stop shipping, sets both against what the change costs in year one, and shows what happens if your assumptions turn out to be wrong.

> ***How to use this template.*** *Allow about forty minutes, and have someone from the delivery team with you. Fill in only the yellow cells. Everything else calculates itself. Where you are unsure, put in the number that makes the case worse. A case that survives pessimistic inputs is the one that gets approved. Do not delete the Sensitivity sheet before the review. It is the sheet that answers the hardest question you will be asked.*

1. Before you open the workbook

Three numbers are worth gathering in advance, because looking them up mid-conversation kills the momentum of the session.

- **Manual regression hours per cycle.** Person-hours to hand-test the regression suite once. Ask the team; they will know, and their figure is better than an estimate.
- **Escaped defects last quarter.** From the incident or ticket system, counted over a full quarter rather than a good month.
- **Blended loaded hourly cost of a tester.** From finance, not from a market rate. Salary plus employment cost over working hours.

> Write the date on the baseline. Six months from now it is the only evidence of where you started, and it is the thing teams most often wish they had captured.

2. Sheet by sheet

Sheet	What it is for
Start here	Orientation. Read it once and move on.
1. Your baseline	What testing costs you today. None of it comes from Studio - it comes from your release process and your ticket system.
2. Assumptions	The three drivers that decide the answer, plus the cost of the tool. This is where the argument actually happens.
3. The business case	The result. Every figure traces back to the two sheets before it.
4. Sensitivity	What happens if you are wrong. Take this into the review.
Notes	Where each number came from, and the challenges to expect.

3. The three assumptions that decide it

Almost all of the disagreement in a review will be about these three. Settle them with the room rather than for the room.

Assumption	Defensible range	What moves it
Share of manual regression that is automatable	60-80%	Exploratory testing, subjective judgment, physical actions and one-time migration checks stay manual on purpose. A lower number here is a sign of a mature view, not a weak case.
Expected reduction in escaped defects	20-40%	Regression automation catches regressions, not new-feature defects. Above 50% you will be asked to justify it, and you should be.
Hours to author one test	1-4 hours	In Studio a test is a described scenario rather than a scripted one. If you have run a pilot, use your measured figure and say so.

> If the case only works at the top of every range, it is not a case yet. Take the middle of each and see whether it still holds.

4. Reading the result

Line	How to read it
Gross annual benefit	Labor plus defects, before what it costs you. Never present this figure on its own - it is the number that makes people suspicious.
Year one net benefit	After the build effort, the maintenance and the license. This is the honest headline.
Payback period	How long before the build effort has paid for itself. Under twelve months is a strong case; under six invites scrutiny, so have the sensitivity sheet ready.
Three-year net present value	Discounted, so it can sit next to other investments on the same slide.

5. The four challenges, and the answers

- **"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 - exploratory testing, or the journeys nobody has time to cover today.
- **"The defect reduction is guesswork."** Partly, which is why the Sensitivity sheet shows the case with *zero* defect reduction. If that line is positive, the defect argument is upside rather than the foundation.
- **"Maintenance will eat it."** The right answer is to measure it rather than argue about it. Maintenance share is one of the five KPIs in *What to Measure*, and it is the number to report monthly.
- **"These numbers came from the vendor."** They did not. Every input is yours, and the workbook has no hidden constants - which is worth saying out loud early in the conversation.

6. For the customer success conversation

If you are walking a customer through this, the useful shape of the session is not a demonstration of the workbook. It is an hour spent finding their real numbers.

1. Ask for the three inputs in section 1 before the call, so the session starts from their data rather than from placeholders.
2. Fill in the baseline together and stop there. Let them react to what their current approach costs before showing any benefit.
3. Set the three assumptions conservatively, out loud, and let them push them *down* if they want. A customer who lowered an assumption themselves defends the result later.
4. Show the Sensitivity sheet before the headline. It changes the register of the conversation from sales to analysis.
5. Leave the file with them. It is theirs to edit, and a calculator that cannot be taken away is not believed.

> The most valuable outcome of the session is often not the number. It is that somebody wrote down what testing currently costs, possibly for the first time.

7. What to do after the decision

A business case nobody revisits teaches an organization that business cases are theater. Put a date in the diary for one quarter out and compare four numbers against the baseline.

- Escaped defects, this quarter versus baseline
- Critical journey coverage, now versus baseline
- Maintenance share of automation effort
- Manual regression hours per release

> **With Functionize Studio:** All four come from places this documentation already covers: Reports for pass rate and trends, the History panel for what changed and when, and your own incident system for escaped defects. See *What to Measure: QE Metrics and KPIs with Studio*.