BYTESIZE STRATEGIES

PROBLEM EXPLORATION GUIDE

The Five Jobs of a Technology Strategy

What your software is supposed to be doing for you, who is supposed to be doing it, and what it costs your mission when no one is.

www.bytesizestrategies.com 917-994-5711 info@bytesizestrategies.com

THE PROBLEM IN PLAIN TERMS

Your organization runs on software. Donor management, finance, HR, program data, email, and the dozen smaller tools that crept in over the years. Together they shape how your staff spend their day and how your leadership sees the organization. They are also one of your largest recurring expenses, and in most midsized nonprofits, the least governed one.

Here is the uncomfortable part. The decisions that govern this software are CIO-level decisions. Which systems to buy. How to evaluate them. When to replace them. How to make them work together. Most organizations your size do not have a CIO, so these decisions get made anyway, by whoever is closest to the problem, under deadline pressure, usually with the vendor in the room.

That is not a criticism of your team. Executive directors, COOs, CFOs, and development officers are exceptionally capable people. It is a description of a structural gap, and the gap has a shape.

There are five jobs that make up a real technology strategy. When they get done, your software serves the mission. When they do not, the software starts to run the organization instead. None of the five is technical. You don’t need to be technical to recognize whether they’re happening. You only need to know what they are.

> ***"Software runs your organization. The question is whether anyone is running the software."***

THE 5 JOBS IN BRIEF

- Know what you have
- Write requirements before you shop
- Manage your vendors
- Build a roadmap your leadership team and board can read
- Treat go-live as the start

JOB 1: KNOW WHAT YOU HAVE

Every application your organization pays for should live on one list: name, cost, who owns it, when it renews, and what it was bought to do. Most organizations cannot produce that list. Software gets purchased department by department, expensed on different cards, renewed automatically, and never reconciled against anything else.

The cost of not knowing is quiet and constant. You pay for licenses no one uses. You pay for two tools that do the same job because two teams each solved the same problem alone. You find out about a renewal when it hits the bank, never in time to renegotiate. And when you sit down to plan, you are planning around a portfolio you cannot fully see.

A simple inventory changes the conversation. Once everything is on one page, the duplication, the waste, and the gaps become obvious. This is the cheapest, fastest technology strategy work there is, and it is almost always where to start.

> **Ask:** ***Can your leadership team list every application you pay for, with the cost and owner of each?***

JOB 2: WRITE REQUIREMENTS BEFORE YOU SHOP

The vendor demo is the most persuasive and least reliable part of buying software. Demos are built to sell: clean data, an ideal workflow, every feature you hoped for, and none of the edge cases that define your actual work. Walk into one without written requirements and you will end up evaluating the vendor's story instead of your own needs.

Requirements are not a technical document. They are a plain statement of what the system has to do: the funder reports you must produce, the integrations you cannot live without, the workflows your staff actually follow, and the dealbreakers as distinct from the nice-to-haves. Write them down before anyone schedules a demo. Then judge every product against your list, not the vendor's.

Organizations that skip this step buy the best demo. Organizations that do it buy the best fit. The difference shows up two years later, when the system either supports the work or fights it every day.

> **Ask:** ***When you bought your last major system, did you have written requirements before the demos started?***

JOB 3: MANAGE YOUR VENDORS

Once a system is in place, the vendor relationship runs for years. Renewals, price increases, change orders, new modules, support tickets. Every one of those is a negotiation, and every negotiation needs someone on your side of the table who can push back.

Most organizations have no one in that seat. So a change order gets proposed and approved, because who would know whether it should have been in scope. A renewal arrives with a price increase and gets paid, because switching feels harder than absorbing it. The upsell gets framed as a partnership, and with no one to question it, that is exactly how it feels. The vendor sets the terms, the timeline, and the price, and the organization calls it a relationship.

Vendor management is not adversarial. It is the discipline of holding a vendor to what they sold you. A vendor who knows someone on your side reads the contract and tracks the commitments behaves differently from one who knows nobody is watching. That shift alone often pays for the oversight.

> **Ask:** ***When a vendor proposes a change order or a renewal increase, who on your side is positioned to say no? If the answer is nobody, that is the gap.***

JOB 4: BUILD A ROADMAP YOUR LEADERSHIP TEAM AND THE BOARD CAN READ

A project list tells you what is happening this quarter. A roadmap tells you where the organization's technology is going over the next three years, and why. The two get confused constantly. Most nonprofits have the first and call it the second.

A real roadmap connects mission goals to system decisions. It sequences the work: this year we stabilize the CRM, next year we replace the finance system because the current one cannot support the new grant reporting, the year after we integrate the two. It names the dependencies, the decision points, and the milestones a board member can actually track. It turns technology from a series of surprises into a plan leadership signed off on.

The board test is the real test. If your three-year technology direction cannot be explained to a board member in a way they can follow and question, you don’t have a roadmap. You have a backlog, and a backlog cannot earn the funding or the patience that real transformation requires.

> **Ask:** ***Do you have a three-year technology roadmap your leadership team and board could read and follow?***

JOB 5: TREAT GO-LIVE AS THE START

Go-live feels like the finish line. The project ends, the team celebrates, the consultants roll off, and attention moves to the next fire. That instinct is exactly backwards. Go-live is the moment the system has to start earning its cost, and the work of making it earn that cost has barely begun.

Systems drift. Configuration that fit on day one slowly stops fitting as the organization changes. Features you paid for go unused because no one circled back to turn them on. Staff who learned the system at launch leave, and the people who replace them learn the workarounds instead. A platform that was a good decision at purchase quietly becomes an underused, half-trusted system three years later, not because the software failed, but because nobody owned it after go-live.

The fix is ongoing ownership: someone watching adoption, retiring what is not working, expanding what is, and keeping the system aligned with where the organization is now rather than where it was at launch. Value creation is the long part. It doesn’t happen by itself.

> **Ask:** ***For systems implemented more than a year ago, can you name what has been optimized, retired, or expanded since go-live?***

WHO DOES THE FIVE JOBS?

Notice what these five jobs have in common. None of them is about writing code or configuring software. They are leadership jobs: knowing, deciding, negotiating, planning, and owning. They are the work a CIO does, separated from the technical execution a CIO oversees.

That separation is the opening for midsized organizations. You don’t need a full-time CIO to get the five jobs done. You need senior judgment applied to them on a schedule that matches your size. That is what a fractional CIO is for: someone who holds these five jobs as their responsibility, a few days a month, so the work actually happens instead of falling to whoever has time and the vendor across the table.

The cost of leaving the jobs undone is rarely a single visible failure. It is the slow accumulation: the duplicate licenses, the wrong-fit system, the renewal nobody caught, the roadmap that never existed, the platform running at half its value. Add it up across a portfolio over a few years and it dwarfs the cost of having someone own the work.

THE ASSESSMENT

Five questions, one per job. Answer each honestly. Count a yes only when you are confident the work is actually getting done, not just that someone could do it if asked.

1. Can your leadership team list every application you pay for, with the cost and owner of each?
2. When you bought your last major system, did you have written requirements before the demos started?
3. When a vendor proposes a change order or a renewal increase, who on your side is positioned to say no?
4. Do you have a three-year technology roadmap your leadership team and board could read and follow?
5. For systems implemented more than a year ago, can you name what has been optimized, retired, or expanded since go-live?

Scoring

4 to 5 yes	**You are doing the work.** Most organizations at your scale are not. Keep going, and protect the discipline as you grow.
2 to 3 yes	**Real technology strategy work is waiting.** The longer it waits, the more it costs. Pick the weakest job and start there.
0 to 1 yes	**Your technology is running you.** The fastest way out is to start with Job One, and put one person in charge of the rest.

You don't need a full-time CIO to do this work. You need someone doing it.

A PRACTICAL NEXT STEP

Send this page to your leadership team and ask each of them to answer the five questions on their own. The disagreement is the useful part. Where your team's answers differ is usually where the gaps are hiding.

Then book a 30-minute call to walk through the answers together. No pitch. You will leave with a clear read on which of the five jobs are getting done, which are not, and what the first move should be.

Book a conversation: bytesizestrategies.com | info@bytesizestrategies.com

About the Author

Rori Gitlin is a fractional CIO and technology strategy consultant with over two decades of experience leading enterprise technology for nonprofit and mission-driven organizations. She has guided technology portfolios across donor management, finance, HR, and program systems, from selection through implementation and the long work that follows go-live. Rori is known for translating complex technology decisions into clear strategic choices, and for being the person in the room who asks the questions a vendor would rather not answer.