SAMPLE

The Digital Automation Business

Automate Your Way to More Income and Less Busywork

Aiman Sterling

Free sample: Introduction & Chapter 1

putrabooks.github.io

Buy the full book:

Ko-fi

Payhip

Contents

Disclaimer

Introduction: The Business of Saved Hours

What Automation Actually Is

The Demand — Where the Manual Work Lives

The Automation Consultant's Skill Set

Choosing Your Service Model

The Workflow Discovery Method

The Automation Audit

The Build Business

The Maintenance Retainer

Pricing the Saved Hours

Finding Automation Clients

The Time-Saved Portfolio

The Automation Product Ladder

When Automations Break — The Failure Modes

Scaling the Automation Practice

The Human Side — Adoption and Change

Legal, Tax, and Liability

The Specialist's Long Game

Your Ninety Days

Final Thoughts — The Compounding of Systems

Quick Start Checklist

Resources

Digital Income

This free sample contains the Introduction and Chapter 1. The complete book contains the full contents above.

Introduction: The Business of Saved Hours

Every business runs on two kinds of work: the work that makes money and the work that moves information around so the money-making can happen. The second kind — copying data between tools, sending the same emails, updating the same spreadsheets, chasing the same confirmations — is the tax every operation pays for existing. Studies of small-business time use keep finding the same numbers: owners and staff spend a quarter or more of their working hours on tasks that no customer would ever pay for directly, because no customer ever sees them.

The automation business sells those hours back.

It is a strange and wonderful business model when you look at it plainly: you find a task a person does by hand every day, build a system that does it by itself, and charge a fraction of what the recovered hours are worth. The client wins the moment the system switches on — they pay $2,000 for the build and recover $10,000 of annual labor value, forever. No other service business offers arithmetic this clean to its buyers, which is why this book spends real chapters teaching you to show the arithmetic rather than talk about tools.

What the business is not. It is not "knowing Zapier." The tools — the automation platforms, the connectors, the trigger-and-action patterns — are the visible layer and the least valuable part of the skill. The valuable layer is everything around them: finding which workflows are worth automating, mapping them honestly (including the ugly exception cases the client forgot to mention), pricing the work against the recovered value, building it reliably, and — the part almost everyone underestimates — maintaining it when the connected tools change their APIs, their formats, their limits. The automation consultant who masters the surrounding skills outsells and outlasts the tool expert every time, and this book is deliberately structured to spend most of its pages there.

The market's moment. Two forces make this the right decade for this business. First, the tools matured: the modern automation platforms connect thousands of business applications, which means the raw material of this business — the ability to make Tool A talk to Tool B — went from a programmer's skill to a configurable one. Second, the tolerance for manual work collapsed: a workforce that has watched software eat every other inconvenience now notices the copy-paste loop with fresh irritation. The demand signal is everywhere once you know its sound — it's the sentence "we have a spreadsheet, but..." — and the chapters ahead train you to hear it, map it, quote it, and keep it running.

How the book runs. Three movements. Foundations (chapters one through six): what automation is, where the demand lives, the skills that matter, and the discovery method that converts a client's vague complaint into a map you can build from. The business (seven through thirteen): the audit offer, the build engagements, the maintenance retainers, the pricing method that sells the saved hours, and the product ladder that turns repeated builds into assets. The practice (fourteen through twenty): the failure modes and their disciplines, the human side of adoption, the legal questions, the specialist's long game, and the ninety-day plan that starts the whole thing. The book assumes no technical background and no existing clients — just the willingness to learn a method and knock on the doors where the busywork lives.

A note on reading order: chapters two through six are the method's core (the pattern, the demand, the skills, the models, the discovery procedure), and the business chapters that follow assume them the way the pricing chapter assumes the arithmetic. Readers who want the executable summary first can start at chapter nineteen's ninety-day plan and work backwards — the book supports both, and the plan makes more sense with the chapters behind it, which is the honest argument for the slow path.

What Automation Actually Is

Strip away the branding and every automation is the same machine: when this happens, do that — automatically. A new form submission happens; a record is created, a confirmation is sent, a task is assigned. An invoice is paid; the access is granted, the spreadsheet is updated, the thank-you goes out. The whole industry, from the simplest consumer tools to the enterprise platforms, is variations on one pattern: a trigger (the event), an action (the response), and logic (the conditions and data in between). The consultant who internalizes the pattern starts seeing automatable work everywhere; the one who memorizes tools sees automations only where tutorials showed them.

Data sync — the copy-paste loops: information created in one place and needed in another (the form in the email tool that needs to be in the spreadsheet that feeds the report that someone retypes into the invoice). Automations replace these with direct connections, and they are the most common first build because they're visible, measurable, and monotonous — the holy trinity of automatable work. Notifications and sequences — the chase-and-confirm loops: the reminder that has to be sent, the follow-up that has to go out three days later, the alert when something crosses a threshold. Automations replace these with scheduled and event-driven messaging, and they're the second most common because the pain of remembering is felt daily. Decision logic — the "if this, then route it there" work: the lead scoring, the assignment rules, the approval flows. Automations replace these with conditions, and they're the most valuable per hour saved because the work they replace is judgment-adjacent — the client pays more to lose it.

Three things automation genuinely cannot do, stated early so the sales conversations stay honest. It cannot handle the exceptions nobody mapped — every workflow has the weird case (the customer who pays by check, the order that ships to two addresses), and an automation handles the exception by failing silently unless the design plans for it. Chapter six's mapping method and chapter fourteen's failure modes exist for this reason. It cannot fix a broken process — automating a bad workflow produces the same bad outcome, faster. The consultant's real first skill is noticing when the client's request should be answered with "your process is the problem; let's fix that first," and the honest ones build their reputations on exactly that answer. And it cannot maintain itself — the connected tools change (APIs, limits, pricing, field names), and every automation has a maintenance tail. The consultants who pretend otherwise sell one build per client forever; the ones who embrace it (chapter nine) build retainer revenue on the truth.

The business's arithmetic, in one paragraph you'll reuse for years: automation

costs are mostly fixed (the build, once; the platform subscription, monthly), while the hours it saves are recurring (every week, forever). A workflow that saves four hours a week saves 200 hours a year; at even $25/hour of loaded labor cost, that's $5,000 a year from a build that took ten hours to finish. The margin in that sentence is why the business exists — and why the pricing method in chapter ten sells the value of the recovered hours rather than the hours the builder spent. The tools are the how. The arithmetic is the business.

For the reader who wants the pattern made concrete before the chapters deepen it: a small booking automation, told as anatomy. Trigger: the form tool's new submission (the client fills the booking form). Actions: create the record in the database (name, service, date — the fields mapped), add the event to the calendar, send the confirmation email with the details, and notify the office channel. Logic: if the date is within 48 hours, flag it for the manual queue (the exception route); if the service is a first-time client, add the intake follow-up sequence. Failure handling: any error notifies the consultant and the client (the silent-failure prevention). Five pieces — trigger, actions, data, logic, failure handling — and every automation in this book, from the simplest sync to the most complex flow, is those five pieces in different arrangements. Learn the anatomy once; recognize it forever.

End of free sample. The full book is available at putrabooks.github.io