



INVESTPUPPY  
UNVARNISHED



InvestPuppy

# 9 Women Cannot Make a Baby in 1 Month

*For anyone else who's been in **that** room.*

*Serious about outcomes. Not serious about ourselves.*

## About InvestPuppy

InvestPuppy builds Vektor, a systematic portfolio management platform for wealth management professionals.

### What this series is

**Every experienced practitioner thinks it. Almost nobody publishes it: *there has to be a better way.***

We thought it long enough that we went and built one.

Before that, we spent decades in the rooms.

**The same project was failing for the third time. The proposal had arrived with twelve consultants, a competitive day rate, and a Gantt chart that implied all twelve would be productive simultaneously.**

The maths looked right. The outcome was already written.

**It isn't incompetence.**

**It's procurement doing its job perfectly, optimising for the wrong variable.**

Nobody planned it this way. Nobody needed to.

**We got frustrated enough to write it down. Then frustrated enough to build something. These papers are the writing-it-down part.**

*The something we built is a separate conversation.*

Fred Brooks named it in 1975. Fifty years later the same meeting is still happening in the same boardroom. A project is late. The proposal on the table is more people. The logic is intuitive: if one consultant takes twelve months, twelve consultants take one month. The logic is wrong. It has always been wrong. It will be wrong the next time this meeting happens. It fails because it treats knowledge work as a production line, where capacity is fungible and output scales with headcount.

### Some things take the time they take.

A baby takes nine months. Not because the process is inefficient. Because the process is sequential. There are things that cannot begin until other things are complete. A financial platform implementation has the same structure.

### The irreducible sequence.

You can have twenty people standing in the corridor waiting for each handoff. The sequence does not compress. The calendar moves at one day per day regardless of headcount.

The irreducible sequence is not a project management problem. It is a physics problem. You cannot parallelize work that depends on the output of preceding work. Adding people to a sequential process does not accelerate it. It creates a queue.

### The coordination tax.

A team of two requires one communication channel. A team of five requires ten. A team of twelve requires sixty-six. Every person added to a project adds  $n-1$  new communication channels. Each channel requires maintenance: alignment meetings, status updates, decision documentation, context transfers.

The fully-loaded cost of a twelve-person consulting engagement includes the client's internal management burden, the rework generated by miscommunication across sixty-six channels, and the delay caused by decisions that cannot be made until all twelve people are in the same room. This cost does not appear on the procurement spreadsheet. It appears on the project timeline six months later.

### The rework multiplier.

A wrong architectural decision in month one costs one hour to make and approximately four months to undo. The decision that cost an hour to make costs a quarter to undo.

**The experienced practitioner recognises the decision point before it becomes a problem. They have seen the cascade that follows the wrong choice. They know which question to ask in the meeting that has not happened yet.**

### Everyone sees it in week two.

The warning sign is visible early. Nobody says so. Not because they cannot see it. Because saying so requires a conversation that is more uncomfortable than not saying so.

**The problem enters the status report as amber. Week three: amber. Week five: amber. Week seven: amber. Day one of week eight: crisis.**

*Next time, the signal will arrive in week two. The status report will turn amber. The structural incentive has not changed. Neither will the outcome.*

**The procurement committee is not wrong.**

### The procurement committee is not wrong.

A board paper that approves eight junior consultants at rate X demonstrably minimises the visible input cost. The procurement committee is optimising for the metric it can measure. Day rate is measurable. Rework probability is not.

The argument for premium rates is not that senior practitioners work harder. It is that they make fewer decisions that need to be reversed. Every reversed decision in a financial platform implementation carries a multiplier. A single avoided reversal typically costs more than the rate differential for the entire engagement.

### Before the next proposal lands on your desk, two questions.

What is the probability that a decision made in month one will need to be reversed in month five? And what does that reversal cost, fully loaded?

*Nine women cannot make a baby in one month. Some things take the time they take. The only variable you control is whether the person doing the work has seen the path before.*

# The Unvarnished Series

Thirteen field notes on why financial software implementations fail.

Published under our own name. All freely available, no registration required.



## Why Implementations Fail (UNV-01)

The same implementation failures repeat across the industry, unrecorded, because no party benefits from writing them down. This paper names the pattern openly, once, rather than let it repeat quietly again.

## Nobody Owns It (UNV-02)

Ask a vendor who owns an implementation and they say the client. Ask the client and they say the vendor -- both are right, and that shared, undefined ownership is the actual failure mode, not any single decision either side made.

## Complexity as Cover (UNV-03)

A consultant's real product is often the document justifying the advice, not the advice itself -- built to survive the project's failure, not prevent it. Complexity becomes the cover story for that gap.

## Plan Last (UNV-04)

Project plans are written before anyone understands the workflow they claim to schedule, in columns spanning weeks one through thirty-six. Planning first and discovering later gets the sequence backwards.

## The Change Request Trap (UNV-05)

A change request is a formal admission that the original specification was wrong, and also a commercial mechanism for charging separately for what should have been included from the start. Neither party is surprised when it happens.

## Invisible Stakeholders (UNV-06)

Every implementation has a second set of stakeholders who never attended a workshop or reviewed a specification, yet are the ones who have to live with what was built. Their absence from the room doesn't make them absent from the consequences.

## The Hammer, the Tool, and the Methodology (UNV-07)

A tool gets picked up, used for the job it suits, and put down. A methodology requires certification, governance, and a steering committee -- and the industry keeps reaching for the second when it only ever needed the first.

## Life After Live (UNV-08)

Go-live is the final milestone on the Gantt chart, but it's the beginning of the harder problem: an organisation built around a legacy system now has to actually live inside the new one. The project plan ends; the real work doesn't.

## The U in UAT (UNV-09)

UAT sign-off documents carry the right signatures -- from consultants testing scripts written by other consultants, not from anyone who actually uses the system day to day. The U in UAT is the part everyone skips.

## Nine Women (UNV-10) (you are here)

Fred Brooks named it in 1975 and the same boardroom conversation still happens fifty years later: a project is late, so the proposal is more people. The logic feels intuitive and has always been wrong.

## The AI Revolution — Automating Bad Practice at Scale (UNV-11)

AI doesn't fix a broken process -- it executes it faster, at greater scale, with fewer chances for the human hesitation that might have caught the problem. The real question isn't whether an AI implementation will work, but what it will end up working very hard to get wrong.

## Model Bank — the perfect solution for banks that don't exist (UNV-12)

A 'model bank' configuration is a composite of institutions that don't exist, built from someone else's defaults before your institution was ever part of the conversation. Every subsequent customisation is just correcting an assumption that was never really about you.

## The parameter that broke the camel's back (UNV-13)

A platform demo answers every question with yes, right up until one specific parameter turns out to be the one nobody actually tested. This paper is the account of exactly where that line sits, and what breaks on the other side of it.

## Our Better Way

*This is what we built instead.*



"So, what did you do about it?" — a fair question after thirteen papers on what goes wrong. InvestPuppy builds Vektor, a systematic portfolio management platform for wealth management professionals — the direct answer, series by series.

These four papers are the how to the thirteen whys above — not a sales pitch, but a technical account of the specific choices Vektor made where the industry's default pattern would have produced the same failures just described.

- Technical Debt (BW-01)
- Vektor House Implementation (BW-02)
- The Configurability Gap (BW-03)
- AI as Instrument (BW-04)

More at [investpuppy.com](https://investpuppy.com)