INVESTPUPPY UNVARNISHED

InvestPuppy

The invisible stakeholders

*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 project plan had been signed off before anyone had understood the first requirement.

The relationship was too warm for anyone to say so.

It isn't incompetence.

It's the incentive structure doing its job perfectly.

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.

Every financial software implementation has two sets of stakeholders. The first set attended the workshops, reviewed the specification, and signed the project plan. The second set did not attend anything. They were not invited, or were invited once and sent a delegate, or were told the project was being handled by the first set and they would be consulted when relevant. They discovered the system on go-live day. They have been explaining why it does not work for them ever since.

1. Who Is Not in the Room

The workshop that produces the specification is not a representative sample of the people who will use the system. It is a sample of the people who were available, willing, and sufficiently senior to attend. These are not the same people.

The operations director attends. The portfolio analyst who processes thirty transactions a day does not. The head of compliance attends. The compliance officer who produces the monthly regulatory report does not.

> The specification that results from a workshop attended by senior stakeholders will correctly describe the high-level workflow. It will incompletely describe the exceptions, the edge cases, the manual processes, and the reporting requirements that matter most to the people who use the system every day. These are the things that generate post-go-live complaints. They were always present. Nobody asked.

2. The Map Is Not the Stakeholder

The stakeholder map is not the same as the stakeholder. This distinction is where most stakeholder engagement

processes fail.

Stakeholder mapping produces a document. The document lists the stakeholders, their level of interest, their level of influence, and their communication plan. It is a governance artefact. It is not an engagement process.

Listing a stakeholder on a map, sending them a fortnightly summary, and asking them to confirm receipt is not stakeholder engagement. It is stakeholder administration. The two things are not the same, and confusing them is one of the primary reasons invisible stakeholders remain invisible.

3. The Empty Chair

There is a moment in every failed implementation where the invisible stakeholder becomes visible. It is usually about three weeks after go-live. And then someone — a compliance officer, a back-office analyst, a reporting team lead — says: this system does not do the thing I need it to do.

They are not wrong. The system does not do the thing they need it to do, because they were not in the room when the configuration decisions were made.

> *There is a more expensive version of this problem. It does not arrive at go-live. It arrives when regulatory requirements change after the system is live — and the architecture cannot accommodate the change because the compliance function was not involved in the design.*
>
> *By this point the project is closed, the team is disbanded, and the cost of the remediation is a new project.*
>
> ***The chair had their name on it. It was empty when the room was full.***

4. Finding the Invisible Stakeholders

The invisible stakeholder is not hard to find. They are the person who currently does, manually or in a legacy system, the thing the new system is supposed to replace. The question that finds them is not 'who is responsible for this function?' It is 'who does this work?'

Three things that surface invisible stakeholders before go-live:

One. A workflow walkthrough conducted with the people who do the work, not the people who manage it — specifically including the exception-handling processes that senior stakeholders may not know exist.

Two. A reporting requirement sign-off process that includes the recipients of every report the system will produce, not only the commissioners.

Three. A UAT process that requires sign-off from every function that will use the system in production. Sign-off by a project team member on behalf of a function is not sign-off.

The empty chair was always there.

It had a name on it.

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) (you are here)
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)
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.

VEKTOR by INVESTPUPPY

"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