



**INVESTPUPPY
UNVARNISHED**



InvestPuppy

9 Women Cannot Make a Baby in 1 Month

Serious about outcomes. Not serious about ourselves.

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.

Also in this series

IP-UNV-01
Why financial software implementations fail

IP-UNV-02
Nobody owns it

IP-UNV-03
Complexity as cover: The consultant layer

IP-UNV-04
Plan last — discovery before configuration

IP-UNV-05
The change request trap

IP-UNV-06
The invisible stakeholders

IP-UNV-07
A hammer is a tool, not a methodology

IP-UNV-08
Life after live

IP-UNV-09
The U in UAT — kind of important...

**IP-UNV-10
9 Women Cannot Make a Baby in 1 Month**

IP-UNV-11
The AI revolution — automating bad practice at scale

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

IP-UNV-13
The parameter that broke the camel's back