INVESTPUPPY UNVARNISHED

InvestPuppy

Plan last — discovery before configuration

Serious about outcomes. Not serious about ourselves.

The project plan is almost always the first formal document produced in a financial software implementation. It describes, in columns, what will happen in weeks one through thirty-six. It is produced before anyone has understood the workflow it is meant to govern. This is not a planning error. It is a procurement requirement. The client asked for a plan. The vendor produced one. The project was signed.

1. The Sequence Problem

There is a correct sequence for implementing financial software. It is: discover, then design, then configure, then test, then deliver. Most implementations do not follow this sequence. Most implementations follow a different one: plan, then configure, then discover, then redesign, then raise a change request, then extend the timeline, then retest, then deliver late.

The second sequence is not produced by incompetence. It is produced by a procurement process that requires a project plan before a discovery process has been completed. The client needs a timeline and a cost before they can approve the budget. The vendor needs a signed contract before they can begin the work. The project plan is the document that connects these two requirements.

It is, necessarily, a work of fiction.

2. What Discovery Actually Is

Discovery is not a workshop. It is not a series of stakeholder interviews documented in a requirements spreadsheet. It is not a two-day session in a hotel conference room that produces a one-hundred-slide deck and a 'discovery complete' milestone.

Discovery is the process of understanding, in sufficient detail, how the organisation currently works — including the workarounds, the exceptions, the manual processes, the data quality problems, and the things that are done by one person who has been doing them for eleven years and has never written them down.

> The organisation that knows exactly what it needs from a new system does not need a discovery process. That organisation does not exist. Every organisation that has implemented financial software has discovered, during or after the implementation, requirements it did not know it had. Discovery does not create these requirements. It finds them before the configuration does.

3. The Configuration Trap

Configuration begins before discovery is complete because the project plan requires it to. Week four is the start of configuration. It says so in the plan. The discovery findings arrive in week three. Some of them are incompatible with the configuration decisions already made in weeks one and two.

This is not a project management failure. It is the project plan doing exactly what it was designed to do — providing a structure that allows the project to proceed before the information needed to proceed correctly is available.

A system configured before the workflow is understood is not a configured system. It is a system configured for the workflow that was assumed. The distance between the assumed workflow and the actual workflow is the scope of the rework.

Every implementation has this distance. Most implementations discover it in production.

The board should have been filled before the first screen was touched.

4. The Correct First Artefact

The correct first artefact in a financial software implementation is not a project plan. It is a workflow map — an accurate description of how the organisation currently processes the things the new system will process.

Three things a real discovery process produces:

One. A workflow map that the operations team recognises as accurate — not the process as it was designed, but the process as it is actually run.

Two. A documented list of exceptions, workarounds, and edge cases that the system will need to handle, provided by the people who handle them daily.

Three. A set of configuration decisions made before configuration begins, with the rationale recorded — so that when assumptions prove incorrect, the correction is made against a documented baseline, not against memory.

Plan when you know what you are planning.

Not before.

Also in this series