



**INVESTPUPPY
UNVARNISHED**

InvestPuppy

**The U in UAT — kind of
important...**

Serious about outcomes. Not serious about ourselves.

The UAT sign-off document has the right signatures on it. The signatories were not users of the system. They were consultants working from test scripts written by other consultants, testing a configuration reviewed by a project team that had not spoken to the people who would use it in production. The document says PASS. The system goes live. Three weeks later, the people who use it every day explain, at length, why it does not.

1. What the U Is For

User Acceptance Testing. Three words. Each one is load-bearing.

Testing means that something is being evaluated against a standard. *Acceptance* means that the evaluation results in a judgment — this works, or it does not. *User* means that the person making that judgment is the person who will use the system in production, on real transactions, under real conditions, with real consequences for getting it wrong.

Remove the user from User Acceptance Testing and you have acceptance testing. Which is a different thing. A system can pass acceptance testing and fail UAT. This happens. It is, in fact, one of the most common outcomes in financial software implementation.

2. How the Outsourcing Happens

The decision to outsource UAT is rarely announced as a decision. It arrives as a resource allocation, a timeline pressure, or a project management convenience.

The users are busy. The system is going live in six weeks. The project team has UAT test scripts. The consulting firm has testers. The testers are available. The users are not. The project manager makes a practical decision: the consultants will run the UAT.

The users are busy because the organisation has not protected their time for UAT. The organisation has not protected their time for UAT because UAT has been treated as a project team activity rather than an operational activity. The resourcing constraint that makes consultant-led UAT feel necessary is a planning failure, not a fixed condition.

3. What the Consultant's UAT Actually Tests

A consultant running UAT with a test script is not testing whether the system works. They are testing whether the system produces the outputs described in the test script, given the inputs described in the test script, in the sequence described in the test script.

This is a narrower question. It excludes, by design, everything that is not in the test script. It excludes the transaction that occurs twice a month. It excludes the workflow exception that the operations team handles manually. It excludes the report that the risk team runs every Friday.

The test script was written from the specification. The specification was written from the workshops. The workshops were attended by the stakeholders who were available.

The users who were not available did not attend the workshops, did not shape the specification, and did not write the test scripts. A consultant running those test scripts is not testing whether the system works for the users. They are testing whether the system is consistent with a document that the users had limited input into.

The U was removed long before the UAT phase began.

4. What Real UAT Looks Like

Real UAT is operationally inconvenient. It requires the users to stop doing their jobs for long enough to test the system that will change their jobs. It requires test scripts that were written with user input, not only from the specification. And it requires someone with the authority

to say, without commercial consequence, that the system is not ready.

Three things that make UAT real rather than documentary:

One. Users — the actual people who will use the system in production — run the tests. Not a representative, not a delegate, not a consultant with a test script.

Two. The test scripts include the transactions that are not in the specification. The exceptions, the edge cases, the processes the users know exist and the project team does not.

Three. The sign-off authority rests with the users, not with the project team. A UAT phase in which the project manager can override a user's rejection of a test case is not a UAT phase. It is a documentation exercise.

User Acceptance Testing.

The word 'user' is not decorative.

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