



SAP User Acceptance Test Plan

Business sign-off for an S/4HANA transformation

UAT is where the business decides whether it can do its job on the new system. In an S/4HANA program it carries an extra weight: for most users this is their first extended contact with Fiori, and their verdict is shaped as much by whether they can find things as by whether the configuration is correct.

How to use this template. Everything in italics is guidance and should be deleted once you have replaced it. Square brackets mark a value to fill in. These four documents are designed to be used together: the strategy sets the rules, and the three plans execute them at each level. Where your program uses different level names, keep your names and map them in section 1 rather than renaming everything.

1. Purpose and boundaries

UAT is	UAT is not
Business confirmation that the process works and is usable	A second run of SIT
Executed by people who do the job today	Executed by the SI or the test team on their behalf
Against realistic data, volumes and roles	Against a handful of clean demo records
A decision: accept, accept with conditions, or reject	A phase that stops when the date arrives

If your UAT is finding functional defects in volume, SIT exited too early. Record that as a program finding rather than absorbing it.

2. Entry criteria

- SIT exit criteria met and signed.
- No open severity 1 or 2 defects.
- Regression suite green on the UAT build.
- UAT environment refreshed with realistic master data.
- Roles and authorizations assigned to the actual test users.
- Named business participants confirmed, with time formally allocated and backfilled.

UAT environment

Build / transport level

Data refreshed on

Regression result and date

3. Participants

The most common cause of a failed SAP UAT is participants who were volunteered rather than released from their day job.

Business area	Named participant	Role(s) assigned	Days allocated	Backfilled by
[Sales]	[]	[Z_SD_CLERK]	[]	[]
[Finance]				
[Warehouse]				

4. Scenarios

Write in the language of the job. "Raise a credit note for a customer who returned two of five items" — not "test VA01 variant 3".

#	Scenario	Business area	Role used	Acceptance criteria	Result
1	[]	[]	[]	[]	[]
2					

Cover at minimum: every critical process from the strategy, one deliberately awkward case per area, period-end activities if they fall in the window, and at least one scenario per role that confirms a user **cannot** do something they should not.

5. Fiori usability feedback

Capture this separately from defects. It is the feedback most likely to be dismissed as noise and most likely to drive adoption problems after go-live.

Observation	Area	Impact on adoption (H/M/L)	Action	Owner
[cannot find the app for X from the launchpad]	[]	[H]	[launchpad grouping / training]	[]

6. Defects and change requests

Type	Meaning	Route
Defect	Does not match the agreed design	Fix before go-live
Change request	Matches the design, and the design is wrong	Backlog — does not block unless critical
Training or usability	Works as designed; the user could not complete it unaided	Training, launchpad or documentation — not a code fix

Separating the third category is what keeps a go-live decision honest. It is also, in most S/4HANA programs, the largest of the three.

7. Sign-off

Decision	Meaning
Accept	Go-live proceeds
Accept with conditions	Proceeds; listed items fixed by an agreed date in hypercare
Reject	Does not proceed; re-entry criteria stated

Decision

Business process owner / date

Program lead / date

Conditions

Hypercare owner

8. Turning UAT into regression

Do this before the participants go back to their day jobs and the knowledge disperses.

Every UAT scenario that will be re-executed at the next wave, the next support pack or the next transport is regression testing wearing a UAT badge. Identify those now.

UAT scenario	Re-run at every wave?	Access route	Deterministic?	Automate?
[]	[yes]	[Fiori]	[yes]	[yes]

Studio drives a **real browser**. That covers **SAP Fiori** and **SAP GUI for HTML (Web GUI)**, plus the surrounding web applications in scope — Ariba, SuccessFactors, Concur, supplier and customer portals, and your own custom front ends. It does **not** drive SAP GUI for Windows, which is a desktop client. Classify each in-scope transaction by its access route before you plan automation, because that one column decides what is and is not a candidate.

With Functionize Studio: because a test is written and read in plain language, the business user who signed the scenario off can read the automated version and confirm it still tests what they accepted. In a program where the SI eventually leaves, that is the difference between a regression suite the business owns and one it merely inherits.