CF // REAL PROJECT / REPRODUCIBLE BUG CASE STUDY

When acceptance inputs escape protection

Clatterforge itself. Two confirmed onboarding defects, one shipped fix, fresh before-and-after execution.

Clatterforge protects acceptance inputs so a coding worker cannot silently change the evidence used to assess its work. A real onboarding bug weakened that boundary: tracked image and lock-file fixtures disappeared from the protected list.

Case	Original behavior	Fixed behavior
Tracked test assets	SVG, PNG and lock-file fixtures omitted	All three retained as protected inputs
Test-edit mode with no protected verifier	Accepted: Ok(())	Rejected with an actionable error
Identical new regression checks	0 pass / 2 fail; exit 101	2 pass / 0 fail; exit 0

Why it happened

Test discovery reused a general repository-file exclusion filter. That filter was useful for browsing source files but filtered out acceptance assets. Separately, enabling test edits skipped automatic test protection without first requiring any protected verifier files to remain.

What changed

The production fix removes the unrelated exclusion filter from tracked-test discovery and rejects test-edit mode when the protected verifier list is empty. The same commit also adds an explicit desktop opt-in and validates CLI input. This reproduction directly exercises the two backend protections.

Prepared November 27, 2026. This is a fresh reproduction of an existing Clatterforge fix, not a newly discovered issue or a new Clatterforge-orchestrated agent run. The tiny temporary repositories supply controlled test inputs; the buggy production code is Clatterforge's actual committed Rust implementation.

01 // FAILURE EVIDENCE

The original behavior

Parent: 3f119cbcd7667b047b8c0fe15765249acb83df23

The captured protected list contains only pyproject.toml. All three tracked acceptance fixtures are missing. The second check then observes Ok(()) for a configuration with zero protected verifier files.

```
running 2 tests
test onboard::team_example_regressions::binary_acceptance_fixtures_must_remain_protected ... ACTUAL protected
files: ["pyproject.toml"]

thread 'onboard::team_example_regressions::binary_acceptance_fixtures_must_remain_protected' panicked at
src\onboard.rs:2244:9:
UNPROTECTED acceptance inputs: ["tests/snapshot.svg", "tests/sample.png", "tests/fixture.lock"]
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
FAILED
test onboard::team_example_regressions::test_edit_mode_must_refuse_zero_protected_verifier_files ... ACTUAL
allow_test_edits=true, protected=[]: Ok(())

thread 'onboard::team_example_regressions::test_edit_mode_must_refuse_zero_protected_verifier_files' panicked at
src\onboard.rs:2254:9:
UNSAFE configuration accepted with no protected verifier files
FAILED

failures:

failures:
    onboard::team_example_regressions::binary_acceptance_fixtures_must_remain_protected
    onboard::team_example_regressions::test_edit_mode_must_refuse_zero_protected_verifier_files

test result: FAILED. 0 passed; 2 failed; 0 ignored; 0 measured; 176 filtered out; finished in 0.26s

error: test failed, to rerun pass `--lib`

PROCESS EXIT CODE: 101
```

02 // PRODUCTION CHANGE

A narrow fix at the boundary

Fix: 392230c214b34fe5e6292419b7c296ce9958ce71
Committed November 26, 2026 (America/Chicago)

```
commit 392230c214b34fe5e6292419b7c296ce9958ce71

    :bug: fix(onboarding): preserve binary acceptance fixtures and verifier guards

diff --git a/src/onboard.rs b/src/onboard.rs
index 9bfb8f4..1a8f335 100644
--- a/src/onboard.rs
+++ b/src/onboard.rs
@@ -602,12 +602,16 @@ pub fn detected_test_files(repo_path: &Path) -> Result<Vec<String>> {
     let paths = git_text(repo_path, &["ls-files"])?;
     Ok(paths
         .lines()
-        .filter(|p| !is_ignored_repo_path(p) && is_acceptance_test_path(p))
+        .filter(|p| is_acceptance_test_path(p))
         .map(str::to_owned)
         .collect())
 }

 fn protect_acceptance_files(req: &mut OnboardRequest) -> Result<()> {
+    ensure!(
+        !req.allow_test_edits || !req.verification_files.is_empty(),
+        "No protected verifier files remain. Disable test edits to keep acceptance inputs protected."
+    );
     if !req.allow_test_edits {
         for path in detected_test_files(&req.repository_path)? {
             if !req
```

The comparison uses Git archive snapshots of the exact parent and fix. An identical added Rust module calls the real private onboarding functions in both snapshots. No working checkout was reset, and no provider or agent was launched.

The screenshot and complete diff are included separately. Git hashes identify the development history; the private development repository is not required to be accessible to readers of this evidence pack.

03 // VERIFIED RESULT

The fixed behavior

```
running 2 tests
test onboard::team_example_regressions::binary_acceptance_fixtures_must_remain_protected ... ACTUAL protected
files: ["pyproject.toml", "tests/fixture.lock", "tests/sample.png", "tests/snapshot.svg"]
ok
test onboard::team_example_regressions::test_edit_mode_must_refuse_zero_protected_verifier_files ... ACTUAL
allow_test_edits=true, protected=[]: Err(No protected verifier files remain. Disable test edits to keep
acceptance inputs protected.)
ok

test result: ok. 2 passed; 0 failed; 0 ignored; 0 measured; 176 filtered out; finished in 0.27s


PROCESS EXIT CODE: 0
```

What this proves

The original code exhibits both reported defects with controlled inputs. The fix retains tracked fixtures and rejects the empty-protection configuration. The two original acceptance-protection tests also pass on the fixed snapshot; their output is included separately.

What this does not measure

This focused replay does not establish desktop interaction, general agent reliability, sandbox isolation, or end-to-end workflow latency. Build-plus-test wall times were 42.384 seconds before and 25.548 seconds after; different cache states make them unsuitable for a speed comparison. Only the focused tests were rerun, not the full product suite.

Included evidence

Three PNG screenshots; this PDF and HTML report; complete before/after logs; original acceptance-test output; production patch; identical added regression module; results JSON; SHA-256 manifest.

Screenshots show actual captured output and the actual patch in a readable evidence viewer. They are not simulated product UI. Host-specific paths and author metadata are removed for publication; test outcomes are unchanged.