FlowX.AI

TECHNICAL PAPER

RECAST · MAINFRAME MIGRATION, CHECKED RECORD BY RECORD AGAINST THE LEGACY PROGRAM

# Change the Rule in the Documentation, Not the Code

*A pre-registered test of COBOL migrations built by a coding agent through FlowX Platform MCP from agent-written documentation alone, run side by side with the legacy programs of AWS CardDemo and compared on every output record.*

**Bogdan Răduță**
Head of Research, FlowX.AI
bogdan.raduta@flowx.ai

FlowX.AI · Applied Research
November 2026 · Version 1.0 · doi.org/10.5281/zenodo.24223975

## SUMMARY

**Scope.** This paper reports one builder, FlowX Platform MCP. The pre-registered protocol also registered a second builder, the FlowX Developer Agentic Harness, an earlier in-product builder; its arms are summarised in Appendix A so that every registered arm is accounted for, and its per-build results are in the repository. Nothing is pooled across the two.

**What we did.** We took AWS CardDemo, open source (Apache-2.0), a sample credit-card mainframe application. One AI read the COBOL and wrote documentation (a wiki). A second AI, a coding agent working through FlowX Platform MCP and given only that documentation, rebuilt the programs on the FlowX.AI platform. We ran old and new on the same data and compared every output record: first the four nightly batch programs, then four online screens, then the whole base application. The protocol was timestamped before any build was scored. The study was then amended. Each amendment is dated, and §1 says which were timestamped before their builds and which rest on git commits.

**Change a business rule by editing the documentation.** We changed one rule in the wiki, and made the same change in the COBOL to get the reference answer. CR2 is the lead rule: a purchase is over limit only beyond the credit limit plus 10%. We know of no public source that proposes it; we kept no search log for this rule.

- **CR2:** the rebuild matched the edited legacy program on every record in 3 of 3 builds. A second rule, CR1 (round interest to the cent), scored 98.99–99.03% in 3 of 3 builds; the one record that differs is a legacy defect the builds do not copy.
- **Six more rules, across all four programs:** 16 of 18 builds followed the change and none kept the old rule. The other two differ on an unrelated address line they had flagged in advance.

**How builds ended.** Every batch build completed with no retry. One screens build was cut off by the account's usage limit and is scored as it stands.

**Disguised, the documentation came within one record of the source code.** On a renamed copy of the interest program, builds from the detailed wiki scored within 1 point of builds given the disguised COBOL (0.996 points; on one set of five, 0.01 over the limit).

**Screens.** FlowX Platform MCP built four online screens three times, scored against the original screens on a stand-in for the mainframe's online system. As registered, H15 (the payment's file state) is not refuted and H16–H19 (screen content, account-view data, browser behaviour, end to end) are refuted. The screens differ from the original.

**The whole base application (A32).** FlowX Platform MCP built all 17 online programs, the date utility and 11 batch programs three times from one wiki of the whole application. The money files were written as legacy wrote them in 3 of 3 builds (HF3 holds). The other six hypotheses (every screen reachable, screens, list paging, browser, end to end, export and import) are refuted in 3 of 3 builds, so the target claim, batch to working screens both checked against the original, is not stated.

**What we do not claim.** The scale is small: four programs of 649–1,154 lines, 50 accounts, five test data sets, three builds per condition. CardDemo is a sample application, run on GnuCOBOL, not

on a mainframe. We measure whether a rebuild behaves like the legacy program, not whether either is correct. This is FlowX.AI testing its own platform.

KEYWORDS Legacy modernisation · COBOL · Mainframe migration · Behavioural equivalence · Coding agents · Pre-registration

# 1 What was tested

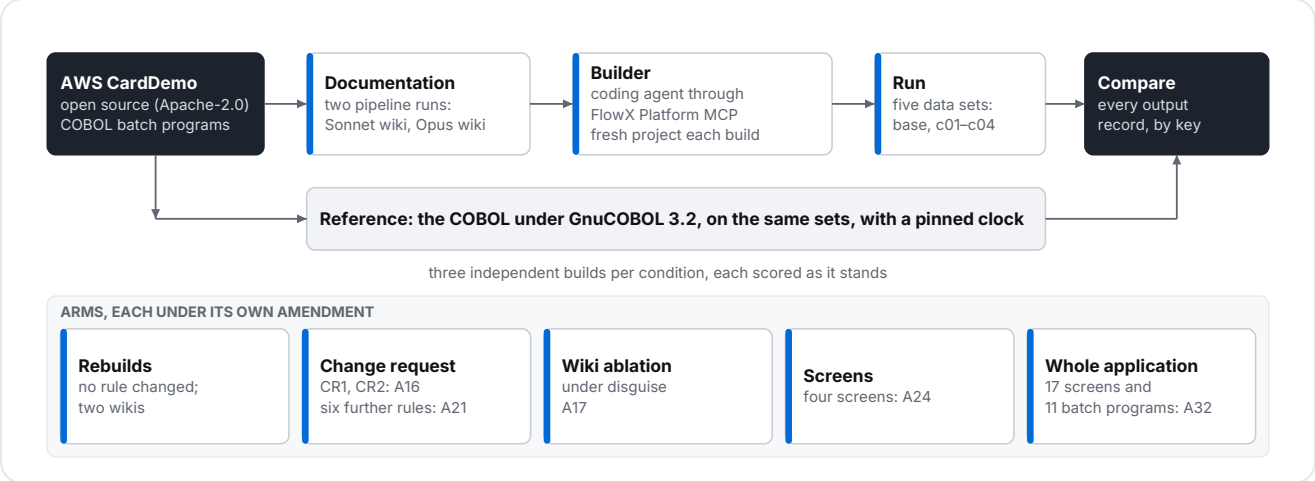


**Figure 1.** Study design and arms reported in this version. One AI documents the COBOL, a second builds only from that documentation, and every output record is compared with the COBOL run on the same data. Diagram: no data plotted.

**Subject and reference.** AWS CardDemo, open source (Apache-2.0):[1] interest calculation (INTCALC), transaction posting (POSTTRAN), statements (CREASTMT) and the daily transaction report (TRANREPT). The reference is the COBOL compiled with GnuCOBOL 3.2[2], unmodified apart from one patch that removes diagnostic displays from the statement program, run with a pinned clock on the shipped data and four derived sets (base, c01–c04). An independent Python re-implementation of each step agrees with it.

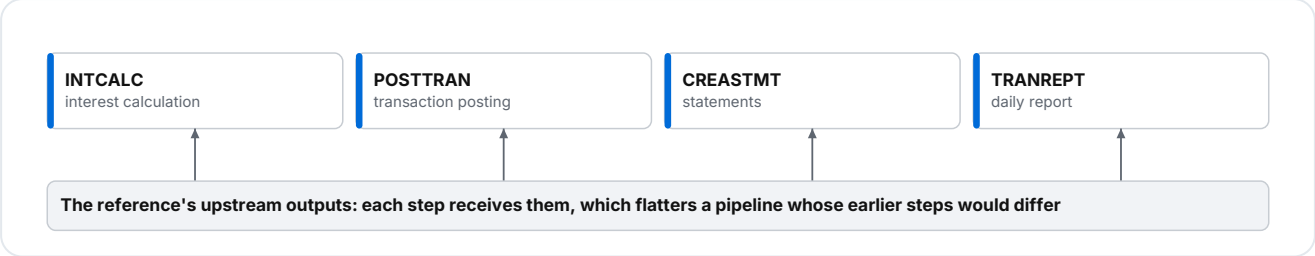


**Figure 2.** The four batch steps tested. Each receives the reference's upstream outputs. Diagram: no data plotted.

**Builder.** A fresh headless coding-agent session (`claude-opus-5-5`) using FlowX Platform MCP and file tools only, with no shell and no web. Each build starts in a fresh project with the same documents and prompt; the prompts are in the repository. Allowed human interventions are infrastructure only. A build that produces no output is scored as it stands; none is dropped after its score is seen. The second registered builder is reported in Appendix A.

**Wikis.** Two runs of a documentation pipeline: the *Sonnet wiki* (detailed) and the *Opus wiki* (thin), named after the model each run used. The names describe the wiki, never the builder.

**Scoring.** One output record is the unit, matched by business key. *Equivalence* is equal records over the union of legacy and candidate keys; *business* equivalence is the primary endpoint. A divergence is *flagged* if the build or the wiki stated that uncertainty in a frozen file before scoring, and *silent* other-

wise. Outcomes are registered at record level. The score measures equivalence with legacy, not correctness: where a build differs, it differs from legacy.

**Pre-registration.** The protocol, scorer and golden outputs were frozen on 2026-10-24, before any build was scored. They were published with this paper, in the repository; the timestamps, not the publication date, carry the order. The package's SHA-256 was stamped with OpenTimestamps[3] and is attested in Bitcoin block 973059. The study did not run exactly as first registered. It was amended (A1–A32); amendments written after a score was seen are labelled exploratory. A1–A4 were committed to git before the first build and are in the package, whose stamp and block came after that build began and before the first score. A5, A6, A7 and A12–A32 were each committed and stamped before the builds they govern. A6b, A8, A9, A10 and A11 have no stamp of their own and rest on git commits; the A12–A14 stamp first covers them, and for A11 it came after every build it governs was scored. The stamps of A16–A32 are attested in Bitcoin blocks 973463–974238, checked against block headers from blockstream.info. A block is an upper bound on a stamp's time; where it falls after the first build, the order rests on the git commit and stamp submission, our own records. That holds for the package and for A12–A14, A19, A20, A23, A24, A25, A27, A28, A30, A31 and A32; for A19 and A24 a build was also scored before the block. For A29 the block falls after its unscored smoke build and 41 seconds before its first counted build. It holds in effect for A7, whose block time is 12 seconds before its first build, too close for block times to settle. For A5, A6, A15, A16–A18, A21, A22 (its first retry) and A26 the block comes before the first build. Instrument changes made after a freeze, and post-hoc corrections, are logged in the repository (`METHODS_LOG.md` and the protocol). The originally registered primary condition was an Agentic Harness arm (Sonnet wiki, INTCALC, hypotheses H1–H4); it is reported in Appendix A.

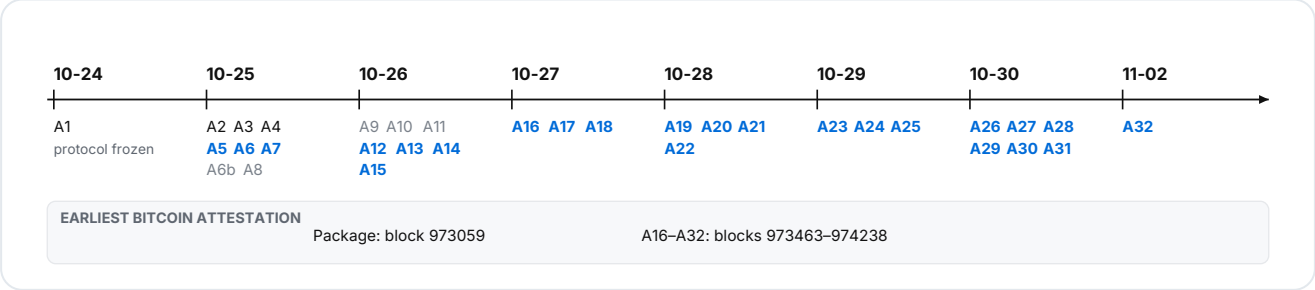


**Figure 3.** Amendments A1–A32, by date. Bold: committed and stamped before the builds it governs. Black: in the package, whose stamp and block came after the first build began and before the first score. Grey (A6b, A8–A11): git commits only, first covered by the A12–A14 stamp. A block is an upper bound on a stamp's time; for the package and for A12–A14, A19, A20, A23, A24, A25, A27, A28, A30, A31 and A32 it falls after the first build. Dates, status and block numbers are parsed from the study's files.

**Prior work.** We found no prior work verifying a change request against the re-run modified legacy program in 3 searches, and none on a memory-controlled documentation ablation against an executed legacy oracle in 3 searches. We do not claim to be first.

# 2 Results

## 2.1 Change request: CR1 and CR2 (A16)

One rule is changed once in the Sonnet wiki and once in the COBOL. CR1 is the contrast: CardDemo's public tracker already proposes it, so memory can supply it. For scale, the unedited legacy program scores 91.1–92.3% against CR2 and 43.4–44.7% against CR1. Business equivalence, %, over base / c01 / c02 / c03 / c04; strict equals business in every cell.

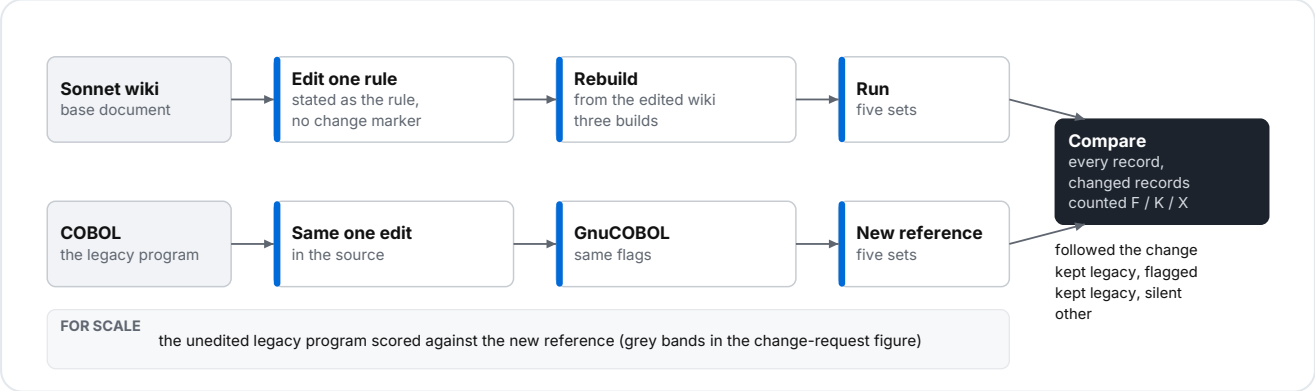


**Figure 4.** How a change request is run and scored: one rule is changed once in the Sonnet wiki and once in the COBOL, and each rebuild is compared with the edited program. F, K and X count changed records that followed the change, kept the legacy value, or differ from both. Diagram: no data plotted.

**Table 1.** CR1 and CR2 (A16): business equivalence over base / c01 / c02 / c03 / c04, %.

| Build | Business % | What differs from legacy | Outcome |
|---|---|---|---|
| CR2 r1–r3 | 100 on every set | nothing | followed the change (3 of 3) |
| CR1 r1–r3 | 99.00 / 99.00 / 98.99 / 99.00 / 99.03 | last account, silent | followed the change (3 of 3) |

The last account is a legacy behavior: the program never rewrites it, and the builds do.

**H11: supported for both rules.** No build followed the legacy rule. All 6 builds completed, with no retry.

## 2.2 Six further rules (A21)

None of the six was found proposed in CardDemo's public tracker or on the web, in 44 searches; nothing stronger is claimed.

**Table 2.** The six further rules (A21), with the unedited legacy program against each new reference.

| CR | Step | Edited rule | Unedited legacy vs the new reference, business % |
|---|---|---|---|
| CR3 | INTCALC | Accounts opened ten or more years before the run date pay the group rate minus 2.00, never below 0 | 70.7–71.8 |
| CR4 | POSTTRAN | At most four purchases per card per run; a fifth that passes the other checks is rejected 104 | 72.0–72.7 |
| CR5 | POSTTRAN | Transactions post in ascending order of amount, credits first; equal amounts keep feed order | 84.4–84.7 |
| CR6 | CREAST-MT | The statement total counts charges only; credits are listed but not subtracted | 50.0 |
| CR7 | TRANREPT | Interest postings are left out of the daily report and its totals | 72.4–73.2 |
| CR8 | CREAST-MT | The statement shows the FICO band in place of the score | 0.0 |

Business %, over the five sets. F / K / X counts the changed records that followed the change, kept the legacy value, or differ from both. The post-hoc field count looks only at the rule's own field; it was defined after A21 was stamped and is never used for H14.

**Table 3.** Six further rules, FlowX Platform MCP: business equivalence, outcome and post-hoc field count.

| CR | Builds | Business % | Registered outcome | What differs from legacy | Field count, post hoc (F / K / X) |
|---|---|---|---|---|---|
| CR3 | r1–r3 | 99.00 / 99.00 / 98.99 / 99.00 / 99.03 | followed ×3 | last account: flagged in r1, silent in r2, r3 | 145 / 0 / 0 each |
| CR4 | r1–r3 | 100 on every set | followed ×3 | nothing | 671 / 0 / 0 each |
| CR5 | r1–r3 | 100 on every set | followed ×3 | nothing | 370 / 0 / 0 each |
| CR6 | r1, r2 | 0 on every set | other ×2 | address line 3 on every statement; the c03 card; flagged | 243 / 0 / 0 each |
| CR6 | r3 | 100 / 100 / 100 / 88 / 100 | followed ×1 | the c03 card on which legacy abends, flagged | 243 / 0 / 0 |
| CR7 | r1–r3 | 99.68 on every set | followed ×3 | grand total, flagged | 496 / 0 / 0 each |
| CR8 | r1–r3 | 100 / 100 / 100 / 88 / 100 | followed ×3 | the c03 card on which legacy abends, flagged | 486 / 0 / 0 each |

**H14: not refuted** (0 of 6 rules). 16 of 18 builds followed the change and no build kept the legacy rule. Part of the match on CR6 r3 and CR8 is recall of the public source: the wiki gives no rule for address line 3.

**Two limits on H14.** Under the registered rubric a wiki TODO counts as a flag, so no CR6 or CR7 build can be "kept legacy, silent": the hypothesis is informative on CR3, CR4, CR5 and CR8 only. And "not refuted" does not say a rule was followed; the outcome column does. All 18 builds completed, with no retry.

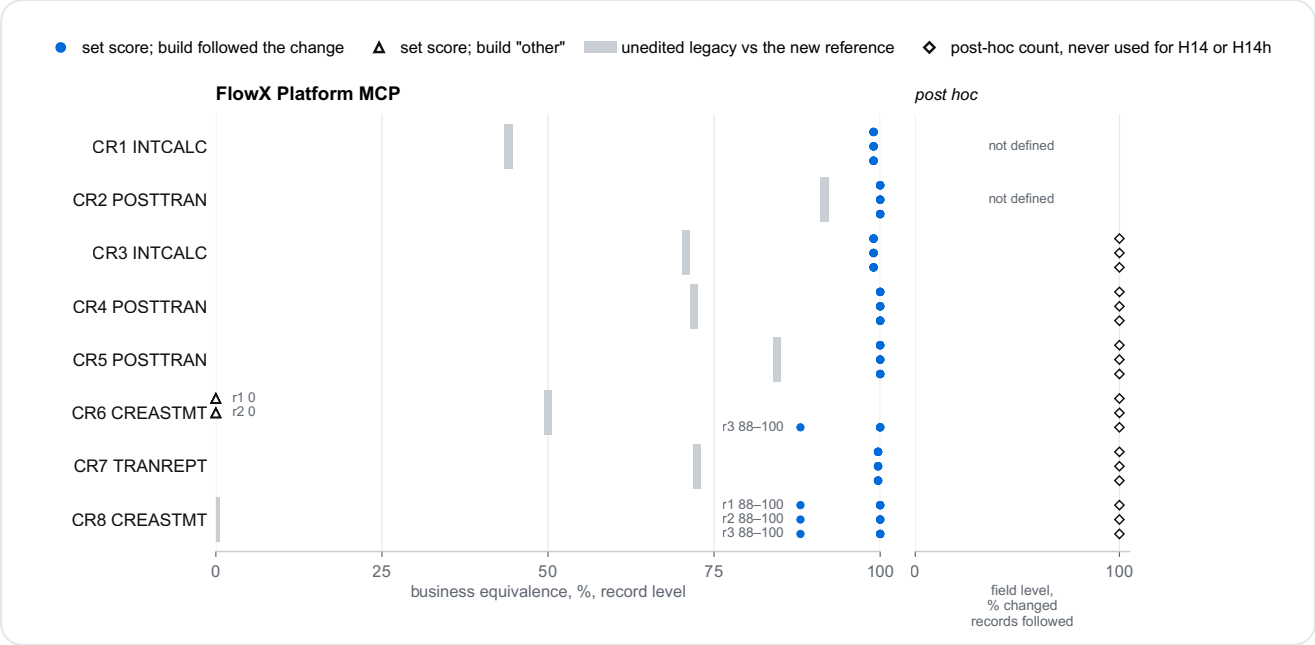


**Figure 5.** Change-request outcomes, CR1–CR8, FlowX Platform MCP, each build shown and never a mean. Each dot is one data set's business equivalence, as registered at record level; rows hold r1, r2, r3 top to bottom. n = 24 builds, five sets each. Grey bands give the unedited legacy program against the new reference. CR6 r1 and r2 stay at 0%: every statement differs from legacy on a flagged address line. Post hoc: the narrow column shows the field-level count, defined after A21 was stamped and never used for H14.

## 2.3 Wiki ablation under disguise (A17)

Every identifier in INTCALC was renamed and five values perturbed, so that recall gives the legacy value. Four arms differ only in the specification: *detailed*, *summary* (stops before the Business Rules section), *no wiki* (record layouts and contract only) and *code* (the disguised COBOL). Three builds per arm. The primary figure is the median over three builds of each build's mean over the five sets.

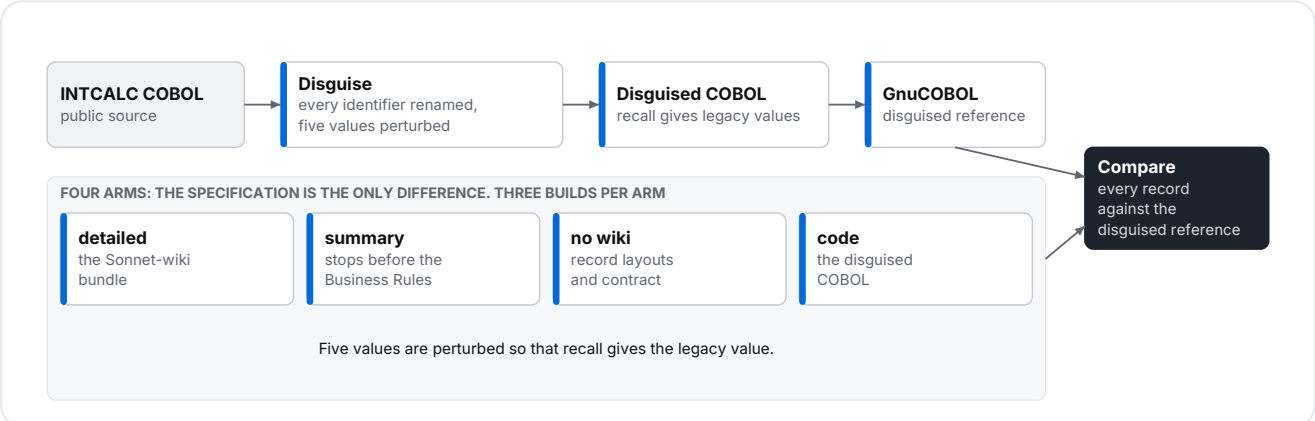


**Figure 6.** The ablation under disguise (A17): on a renamed INTCALC, four arms differ only in the specification. Diagram: no data plotted.

**Table 4.** Wiki ablation under disguise (A17): the median of build means and the build scores on base.

| Arm | Median of build means | Build scores on base |
|---|---|---|
| detailed | 99.004 | 99.00 / 99.00 / 99.00 |
| summary | 99.004 | 99.00 / 99.00 / 2.00 |
| no wiki | 2.396 | 1.00 / 2.00 / 2.00 |
| code | 100.000 | 100 / 100 / 100 |

- **H12 (detailed within 1 point of code): supported at the margin.** The gap is 0.996 points on the pre-registered figure. Per set it is 1.00 / 1.00 / 1.01 / 1.00 / 0.97, so c02 alone is 0.01 over the limit. Both readings are reported.
- **H13 (detailed more than 1 point above summary): not supported** (0.00 points): the summary builds filled the gap from recall of CardDemo.
- **Instrument check (no wiki more than 5 points below detailed): passes** at 96.6 points. The no-wiki builds identified CardDemo and used recalled values, so that gap measures the wiki against recall.
- **Strict scores.** The table gives business equivalence. Strict is lower for the summary builds (49.00 / 49.00 / 1.00 on base).

**How these builds ended.** 12 of 12 completed, with no retry.

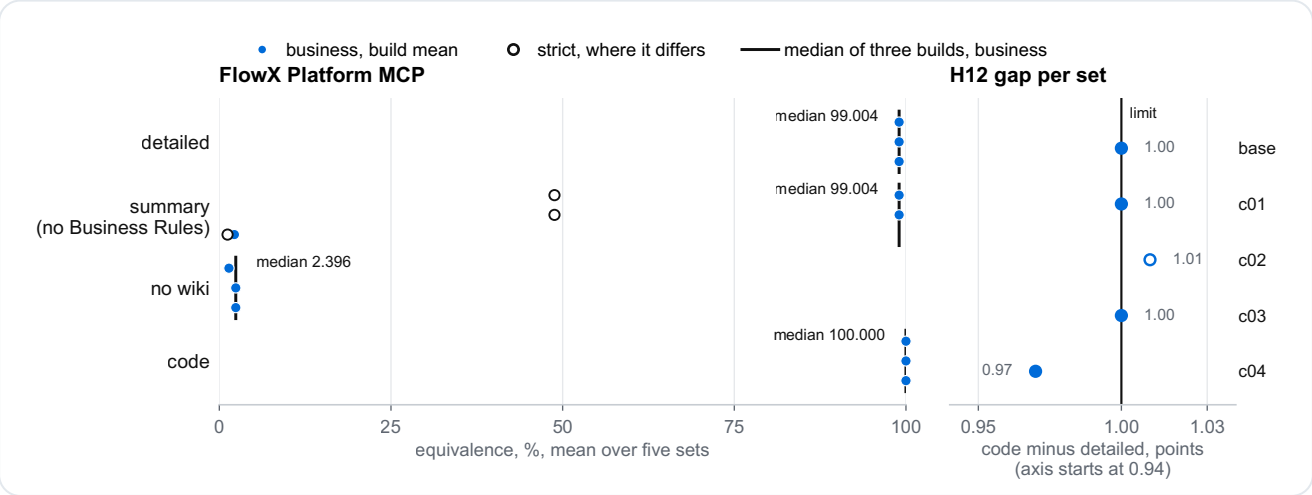


**Figure 7.** Wiki ablation under disguise (A17), FlowX Platform MCP: the score with and without the Business Rules section. Each filled dot is a build's business mean over five sets, each open dot the strict mean where it differs, the tick the pre-registered median of three builds; n = 12 builds. The summary builds filled the gap from recall of CardDemo, and the no-wiki builds used recalled values, so those gaps measure the wiki against recall. Right: H12's gap per set; the open dot is c02, 0.01 over the limit.

## 2.4 Rebuilds from the documentation, no rule changed

24 builds, 24 completed, no retry. Business equivalence, % over the five sets.

**Table 5.** Rebuilds with no rule changed, FlowX Platform MCP: business equivalence, % over the five sets.

| Step | Sonnet wiki, r1–r3 | Opus wiki, r1–r3 |
|---|---|---|
| INTCALC | 98.99–99.03 ×3 (last account: silent in r1, flagged in r2, r3) | r1 100; r2, r3 98.99–99.03 (flagged) |
| POSTTRAN | 100 ×3 | r1 100; r2 84.73–87.37; r3 85.74–87.37 (all flagged) |
| CREASTMT | 100 on four sets; on c03 100 / 100 / 88 (flagged) | 100 on four sets; on c03 88 / 88 / 100 (flagged) |
| TRANREPT | 99.72–99.73 ×3 (grand total, flagged) | 100 ×3 |

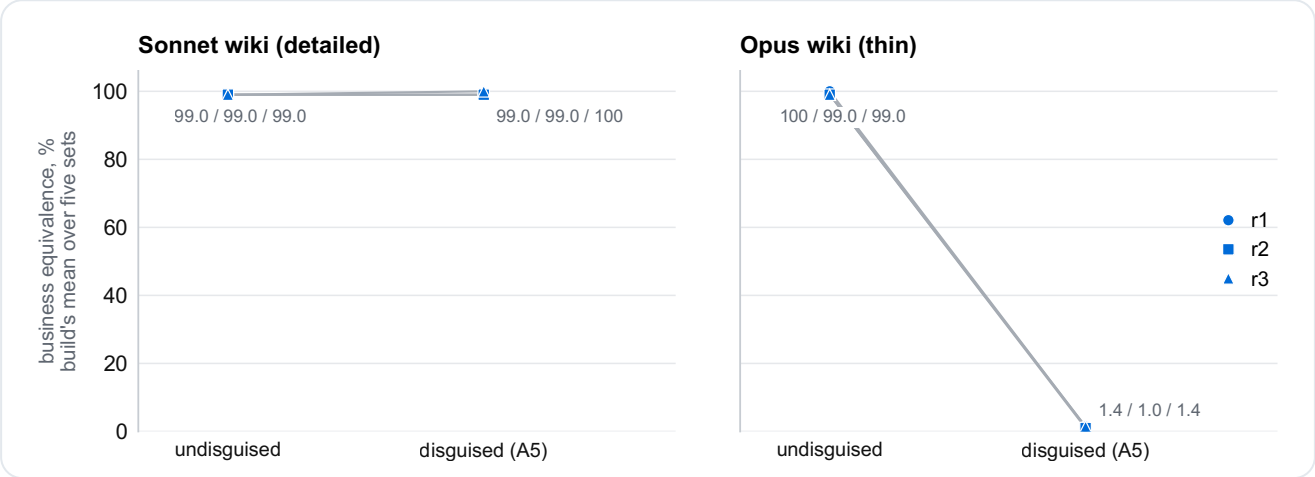


**Figure 8.** Recall control: FlowX Platform MCP on INTCALC, undisguised and under the A5 disguise, three builds per wiki (n = 12). Value labels list r1 / r2 / r3; markers tell the builds apart. Not post hoc.

CREASTMT counts statements, with one overflow statement excluded on c04, and TRANREPT counts report units. The Opus-wiki matches lean on recall of the public source, and the builds say so: under a disguise (A5) the three Opus-wiki INTCALC builds scored 1.4 / 1.0 / 1.4 overall, and the three Sonnet-wiki builds 99.0 / 99.0 / 100.

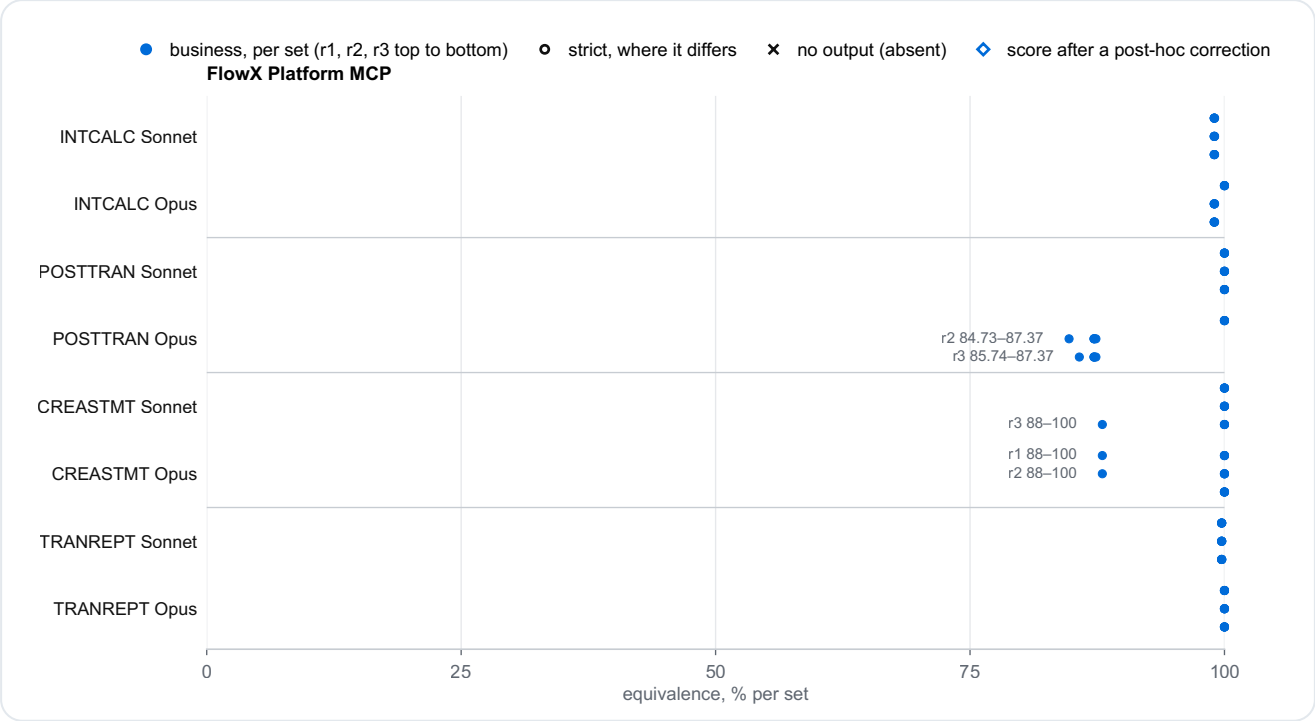


**Figure 9.** Rebuilds with no rule changed, FlowX Platform MCP, per program and wiki. Each filled dot is one set's business equivalence, each open dot its strict score where the two differ by more than half a point; r1, r2, r3 top to bottom. n = 24 builds. The matches on the Opus wiki lean on recall of the public source.

## 2.5 Flagged in advance

Most builds that differed from legacy said so in advance, with limits. In the main arms one build's divergence was silent, and the same last-account divergence is silent in all three CR1 builds and in CR3 r2 and r3. Under the A5 disguise, four of six builds had a silent divergence, so part of the flagged record on public code comes from memory of it. The screens builds have silent units too (§2.6, §2.7). Flags are coarse, and the flag mapper is not independent (§3).

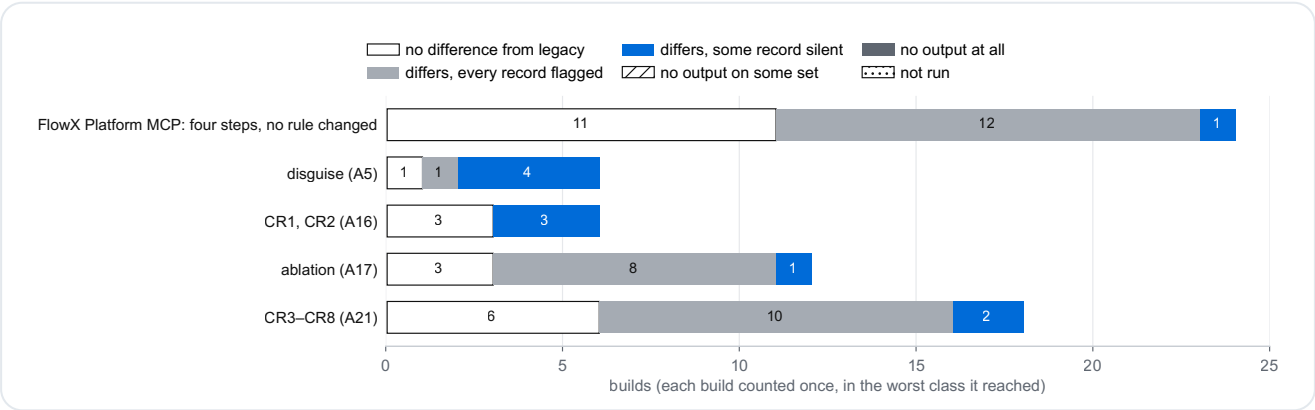


**Figure 10.** Whether builds said so in advance, FlowX Platform MCP. Each build is counted once, from the silent and flagged counts in its report files; a build or set that returned nothing counts as silent and is drawn apart. Flags are coarse, and the flag mapper is not independent (§3). Classes are derived for this figure from the registered counts; no hypothesis uses them. n per row = builds in that arm.

## 2.6 Screens (A24)

The test uses one wiki and four screens: sign-on, main menu, account view, bill payment. Each build ran 17 online cases (16 scored) at workflow level, a browser subset of four cases, and one end-to-end case. A hypothesis is refuted if two or more of the three builds fail it.

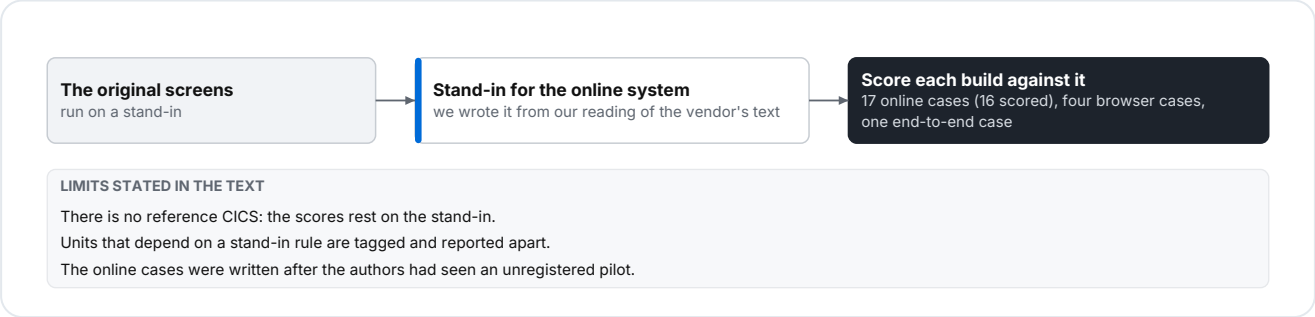


**Figure 11.** The stand-in for the mainframe's online system used by the screens test (A24). Diagram: no data plotted.

### As registered.

**Table 6.** Screens (A24), FlowX Platform MCP, as registered.

| Hypothesis | r1 | r2 | r3 | Verdict |
|---|---|---|---|---|
| H15 payment files, silent units | 0, holds | 0, holds | 0, holds | not refuted |
| H16 core screens, silent units | 23, fails | 171, fails | 113, fails | refuted |
| H17 account view, fields equal | 16 of 29, fails | 12 of 29, fails | 12 of 29, fails | refuted |
| H18 browser subset, steps | 24 of 25, fails | 24 of 25, fails | 24 of 25, fails | refuted |
| H19 end to end | payment differs, fails | payment differs, fails | payment differs, fails | refuted |

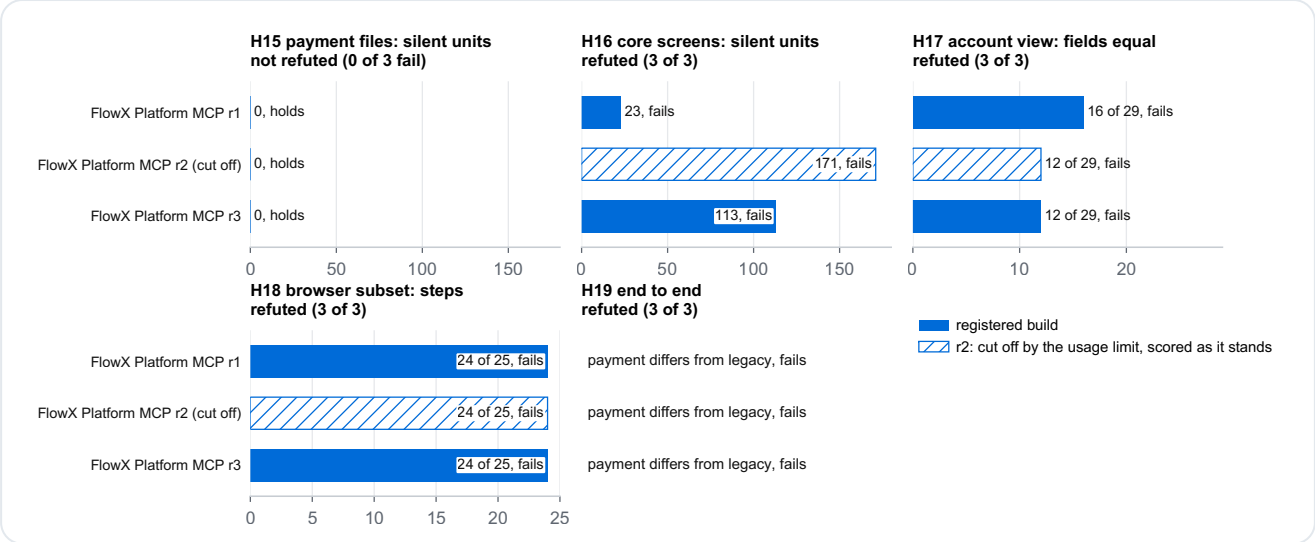


**Figure 12.** Screens (A24), FlowX Platform MCP, per hypothesis and build, scales differing per panel. n = 3 registered builds; r2 was cut off by the account's usage limit and is scored as it stands, with no flags, so every difference in it counts as silent. The exploratory extra build is not counted and not plotted. H15 covers the two registered payment cases; a third, tagged payment case differs in r2 and r3 (next figure).

**Table 7.** Screens (A24), FlowX Platform MCP: descriptive measures, no threshold.

| Descriptive, no threshold | r1 | r2 | r3 |
|---|---|---|---|
| Screen units equal, untagged | 186/547 | 178/533 | 177/533 |
| Screen units equal, tagged (rest on a stand-in rule) | 58/200 | 54/172 | 55/172 |
| File units equal, untagged | 6747/6747 | 6747/6747 | 6747/6747 |
| File units equal, tagged | 8/8 | 6/8 | 6/8 |
| Cost; builder time | $10.25; 18 min | $12.04; 24 min | $13.64; 24 min |

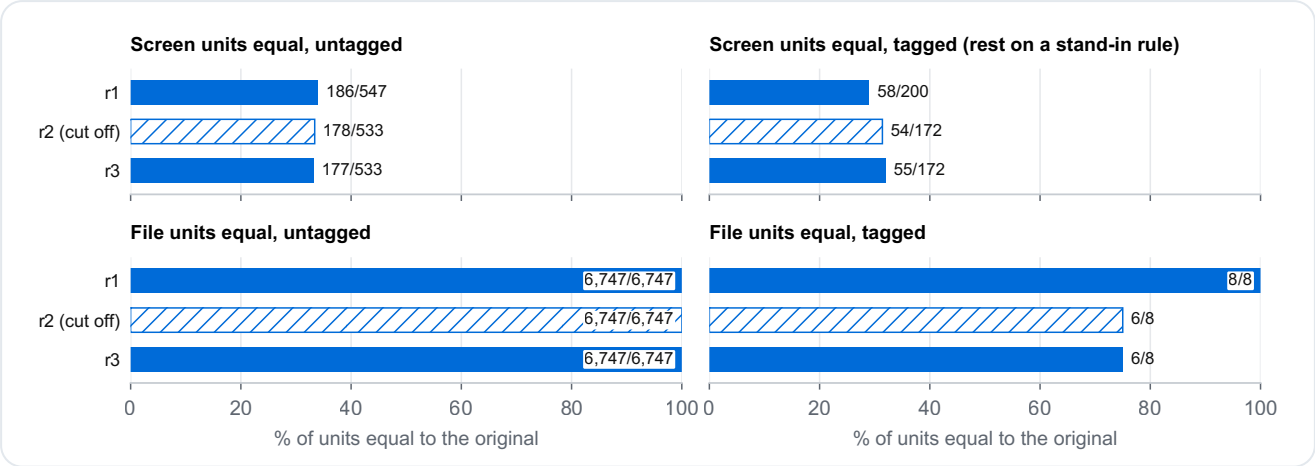


**Figure 13.** Screens (A24), descriptive measures with no threshold: units equal to the original, FlowX Platform MCP, n = 3. r2 is scored as it stands. Tagged units rest on a stand-in rule.

The target claim A24 registered, batch through to working screens, is **not stated**: it needs H15–H19 all to hold. What the data supports is narrower: screens were built and scored against the original, the payment's file state holds on the two registered payment cases (a third, tagged case differs in two builds, unflagged in one), and the screens differ. H19's failure is by construction for this builder (measured): its posting build stamps the wall clock, so the report's July 2022 window drops every row. On H18, all builds fail at the same step on the same four date fields; that platform date rendering is the cause is an inference, not tested.

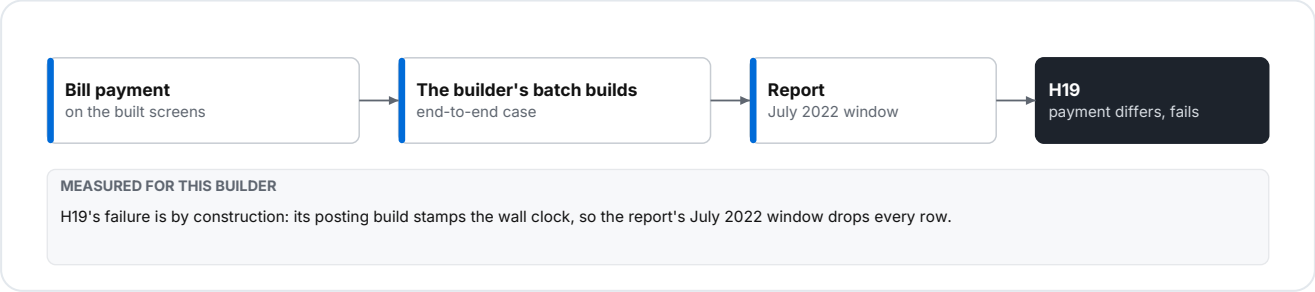


**Figure 14.** The end-to-end case in A24, as far as this version's text describes it. Diagram: no data plotted.

**How these builds ended.** r1 and r3 completed. r2 was cut off by the account's usage limit, ended early and is scored as it stands, with no flags, so every difference in it counts as silent. An exploratory extra build, added after the limit reset, is not counted; with it in r2's place the verdicts do not change.

## 2.7 The whole base application (A32)

**Design.** The rest of the base application, measured as §2.6 measured four screens and §2.1–2.4 the four batch steps: 17 online programs (4 of them in §2.6), the date utility and 11 batch programs (the 4 core steps and 7 others). One wiki of the whole application (234 pages), three builds, `claude-opus-5-5`, each in a fresh project, from answer keys frozen before the amendment: 54 scored screen cases, 23 batch cases, 4 end-to-end pairs and 9 browser cases. Each hypothesis is refuted if two or more of the three builds fail it, decided on units that do not rest on the stand-in's reading of one mainframe date service (R-CEE-PIC8), which are reported in their own column.

**Three contract additions** make A32 a different test from §2.6: a store for job requests, a run date set by the operator as a project parameter before each build started (with where a workflow reads it), and a text on dates on screens. So A32's browser and end-to-end hypotheses are not §2.6's, and A32 is never shown as an improvement on it.

**Table 8.** The whole base application (A32), FlowX Platform MCP, as registered.

| Hypothesis | r1 | r2 | r3 | Verdict |
|---|---|---|---|---|
| HF1 all 17 screens built and reachable | 4 not reachable, 2 differ | 2 not reachable, 1 differs | 3 not reachable, 1 differs | refuted |
| HF2 programs whose screens differ only where flagged | 0 of 17 | 1 of 17 | 1 of 17 | refuted |
| HF3 money files, silent units | 0, holds | 0, holds | 0, holds | **not refuted** |
| HF4 list paging, units equal | 336 of 348 | 336 of 348 | 336 of 348 | refuted |
| HF5 browser cases, all steps as legacy | 5 of 9 | 8 of 9 | 1 of 9 | refuted |
| HF6 end to end, pairs that carry through | 0 of 4 (absent) | 2 of 4 | 2 of 4 | refuted |
| HF7 export and import round trip | fails | fails | fails | refuted |

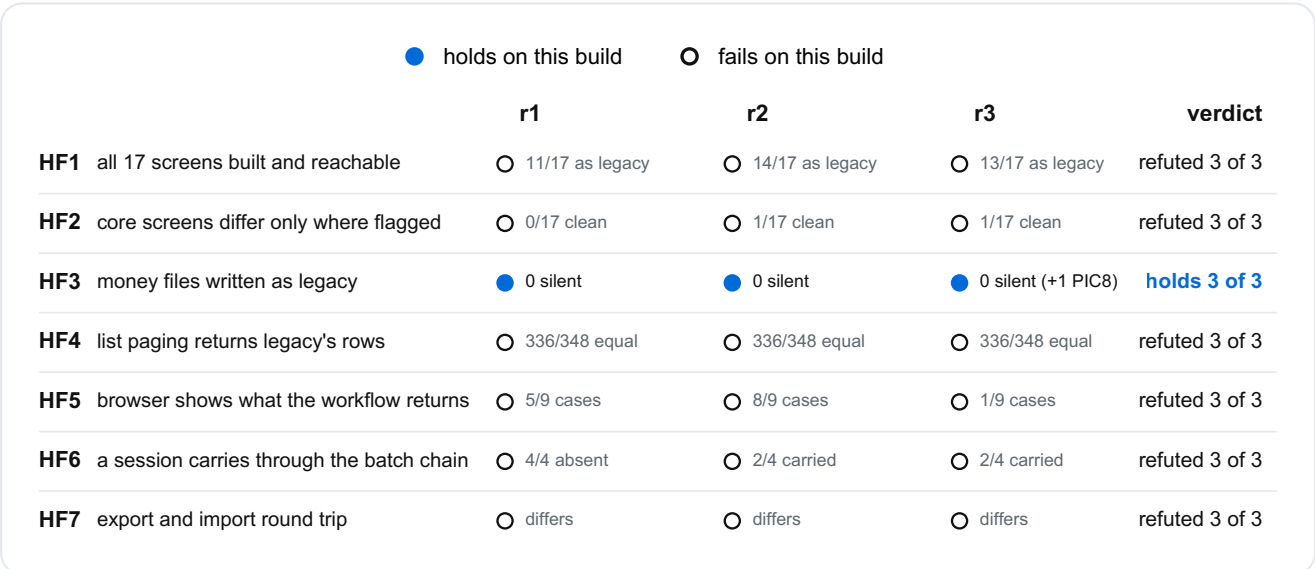


**Figure 15.** The full base application (A32), FlowX Platform MCP: HF1–HF7 per build, as registered; n = 3 builds. A filled dot holds, a ring fails. Cells: HF1 programs reached as legacy; HF2 programs with no silent screen difference; HF3 brackets the units resting on R-CEE-PIC8, which decide nothing. A hypothesis is refuted if at least 2 of 3 builds fail. Only HF3, the money files, holds. The contract adds a store, a run date and date-display text that the screens test (A24) did not carry, so these results are not a comparison with A24. Not post hoc.

**Table 9.** The whole base application (A32): descriptive measures, no threshold.

| Descriptive, no threshold | r1 | r2 | r3 |
|---|---|---|---|
| Screen units equal, untagged | 1,855/3,278 | 1,928/3,254 | 2,102/3,262 |
| File units equal, untagged | 25,493/25,493 | 25,488/25,492 | 25,488/25,493 |
| R-CEE-PIC8 column, units equal | 2,779/3,045 | 2,768/3,045 | 2,770/3,045 |
| Cost; builder time | $37.67; 69 min | $35.62; 75 min | $35.77; 74 min |

**What the failures are (measured).** Two to four of the 17 screens are not reachable in each build. r1's batch workflows compute their outputs and pass none of them on at the end, so every r1 batch output is absent, which is scored as build behaviour. In r2 and r3 the four core steps score as in §2.4, the transaction add and the report request carry through end to end, and the account update and a second report request differ from legacy. The export and import round trip matched 414–416 of 483 and 464 of 477 units.

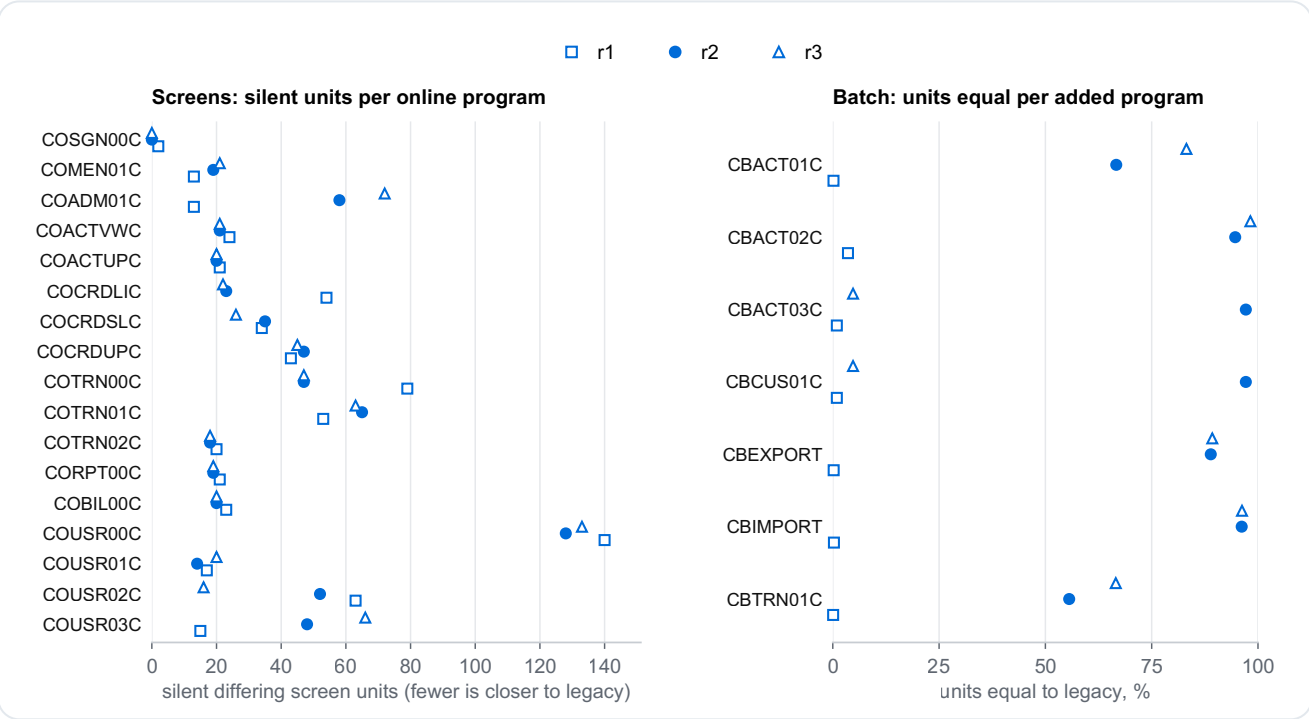


**Figure 16.** The full application per program, FlowX Platform MCP, n = 3 builds; markers tell the builds apart. Left: silent differing screen units per online program (0 is no silent difference). Right: units equal to legacy for the seven added batch programs; five of them have no behaviour in the wiki and carry no hypothesis. r1's batch workflows pass no output on, so it sits near 0. Descriptive; no threshold.

**Instrument, disclosed.** Four instrument fixes (how legacy fields map to web fields, conditional actions, browser handling of menus and choices, and one constant start parameter) were made after the freeze and before any output they first served was executed, each self-tested; answer keys, cases and scorers are unchanged. The browser check ran in the operator's signed-in browser. A post-hoc rerun with a fifth fix, found on r3's registered run, moved r3 from 1 to 5 of 9 browser cases and left r1 and r2 unchanged; it decides nothing, and the registered result stands.

**How these builds ended.** All three completed, with no cut-off or resume. Each launch excluded the operator's global instruction file, and each session's load record shows it was not loaded.

## 2.8 Cost

A FlowX Platform MCP build of one step cost $2.29–5.44 in model usage at list price (median $3.46, 42 builds of the main, A5, A6 and A6b arms) and took 4.9–13.6 minutes (median 7.5). Later builds fall outside these ranges: down to $1.56 and 4.4 minutes, and $10–14 for a screens build. Its 18 builds of the six rules cost $57.00 in all (mean $3.17), 6.7 minutes each on average. The three full-application builds cost $35.62–37.67 each ($109.06 in all) and took 69–75 minutes. The figures exclude generating the wiki, human time and running the workflows.

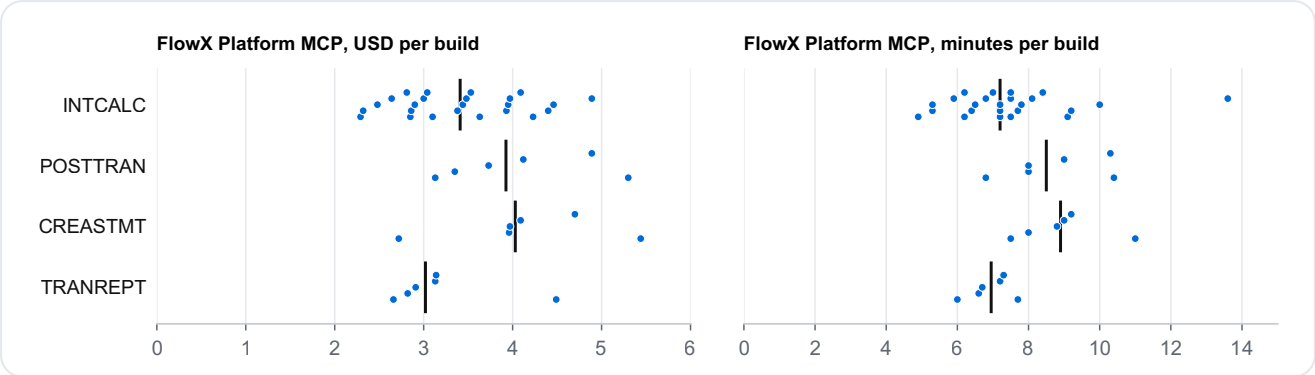


**Figure 17.** Model cost and time per FlowX Platform MCP build, from the run logs. Each dot is a build, the tick is the step's median; n = 42, the main, A5, A6 and A6b arms; later arms fall outside this set (§2.8). Excludes generating the wiki, human time and running the workflows. Axes start at zero.

## 3 Threats to validity

> *We measure whether a rebuild behaves like the legacy program, not whether either is correct.*
>
> RECAST · FROM THE SUMMARY

- **Scale and oracle.** 50 accounts, three builds per condition, description only with no significance tests. GnuCOBOL is not z/OS, and the independent re-implementations share our reading of the source. Each step receives the reference's upstream outputs, which flatters a pipeline whose earlier steps would differ.
- **One builder, one model.** Every build used one coding agent and one model, so the results describe that combination, not FlowX Platform MCP with any agent. The two wikis come from different runs.
- **One of two registered builders.** The protocol registered a second builder, reported in Appendix A. This paper centres on FlowX Platform MCP by the authors' choice, made after both builders' results were known; no FlowX Platform MCP result depends on the second builder.
- **Recall.** CardDemo is public. Recall affects FlowX Platform MCP results on the Opus wiki and part of those on the Sonnet wiki; only INTCALC has a disguise arm.
- **Builder isolation.** Before A32, a FlowX Platform MCP builder is a headless session on the operator's machine, and the launch did not exclude the operator's global instruction file, which was not a registered input. Of the 83 builder transcripts searched, 8 show its content, and that is a lower bound. It holds a context-budget note and a sign-in procedure, with no CardDemo content, but it names the environment and mentions logging interventions in a study run, so a builder could read that it might be in a study. No builder used the procedure. Its content before its last edit on 2026-10-26 is unknown, and builds without it were not tested, so an effect is not excluded. A32's three launches excluded the file, and each session's load record shows it was not loaded.
- **Usage limit.** The same account limit decided where screens r2 ended and that it has no register. A repeat could end elsewhere. r3 ran on usage overage, uninterrupted, and the account login changed once, after every registered build had ended.
- **Online oracle.** There is no reference CICS: the A24 and A32 scores rest on a stand-in we wrote from our reading of the vendor's documentation. Units that depend on a stand-in rule are tagged and reported apart. The online cases were written after the authors had seen an unregistered pilot.
- **CEEDAYS assumption.** A32 is scored on a later version of the online oracle, covering all 17 base screens. Its largest assumption is the 8-byte CEEDAYS call in the account-update program: the vendor's text does not settle what the service does with trailing blanks in the date picture. We assume a valid date is accepted. A third reading is not excluded, and under it every case past a change on that screen would come out differently. The units resting on it are reported in their own column and decide nothing. It was not checked on a real mainframe.
- **A32's instrument and operator steps.** Four instrument fixes came after A32's freeze, before the output they first served was executed; a fifth is post hoc and decides nothing. The run date, the

store loaders and the signed-in browser are the operator's. Three builds from one wiki support description only.

- **Flags.** The flag mapper is Claude, the AI assistant that ran the experiment, under a fixed rubric and before outputs were opened. It is not independent. Per-rule flags are coarse.
- **Change-request leakage.** On CR1 and CR2 the builder can see the bundle's file name; no register mentions a change request.
- **Author affiliation.** The authors build the tools under test. Mitigations are the frozen, timestamped protocol, and keeping every run, flag file and raw output.

## 4 Data and code

Everything needed to check these results is in https://github.com/flowx-ai/carddemo-equivalence-study: the legacy side under GnuCOBOL with its case sets, the frozen answer keys, the wikis and the builder prompts, the scorers and their self-tests, the protocol with every amendment and its OpenTimestamps proofs, and per build the flags, outputs and scores. Re-running a build needs FlowX.AI platform access.

## Appendix A. The second registered builder

The protocol registered a second builder from the start: the FlowX Developer Agentic Harness, an earlier in-product builder, driven through its API with every question answered mechanically (`gpt-6.1-sol`). The originally registered primary condition was one of its arms. It was improved during the study; when a defect halted an arm, the remaining builds ran on a corrected version under a new amendment, stamped before those builds. Its arms are summarised here so that every registered arm is accounted for. Per-build results, flags and outputs are in the repository, with a list of every stop and its effect on the record. Nothing is pooled with FlowX Platform MCP, and the two builders used different models.

**Table 10.** FlowX Developer Agentic Harness: every registered arm, on the version named in each row. Never pooled with FlowX Platform MCP.

| Arm | Version | Outcome |
|---|---|---|
| Four steps, first scored run; primary condition (INTCALC, Sonnet wiki), H1–H4 | first study version | 7 of 24 builds returned output; the primary condition returned no records, so H1, H2 and H4 are not supported and H3 is not assessable there |
| Four steps, run after fixes (A15, exploratory) | A15 | 19 of 24 run (5 not run, by decision); 12 returned output as first decoded, 14 with two labelled post-hoc corrections |
| CR1, CR2 (A16) | A19 | CR1 followed in 3 of 3 (98.99–99.03%); CR2 put in code in 3 of 3, every record matched in 1 of 3 (the other 2 required a run timestamp the test does not supply). H11h supported for CR1; for CR2 not refuted, the per-build criterion met by 1 of 3 |
| Wiki ablation under disguise (A17) | A20 | medians: detailed 99.004, summary 48.812, no wiki 0.996, code 100.000 (one code build left no workflow). H12 supported at the margin; H13 supported |
| Six further rules (A21) | A20, A23, A25 | 18 counted builds: 10 followed the change, 8 "other", none kept legacy silently; H14h not refuted. CR6 and CR8 are 0% and "other" in all 6 builds (a flagged address line on every statement) |
| Four steps on the latest versions (A18 (a), exploratory) | A30, A31 | 24 of 24 returned output (23 completed); 7 builds have a silent divergence at base |
| CR6–CR8 on the latest versions (exploratory) | A28–A31 | the same outcome classes as the counted builds |
| Four screens (A24) | A26–A31 | no screen built; H15h–H18h refuted on 2 of 3 counted builds, both scored absent; H19h not run. Both stops were read as the builds' own and await a ruling; read as defects, they would leave no verdict |
| Whole base application (A32) | – | not run: the bundle (270,305 characters) exceeds what the deployed version accepts (140,000) |

Fifteen stops halted an arm or lost a build; each but the last was followed by a corrected version under a new amendment. Across arms, the second builder returned output less often than FlowX Platform MCP, and where it returned output its divergences were mostly flagged.

## References


- AWS, CardDemo mainframe modernization sample, open source (Apache-2.0). `https://github.com/aws-samples/aws-mainframe-modernization-carddemo`

- GnuCOBOL 3.2.
- OpenTimestamps.

**FlowX.AI**

FlowX.AI builds and orchestrates enterprise-grade AI agents for regulated industries, with a focus on banking, financial services, and insurance. The study behind this paper ran FlowX.AI's own builders on a public sample application (AWS CardDemo, Apache-2.0) under GnuCOBOL; no client code or client data was used.

FlowX.AI Technical Paper · RECAST version 1.0, 3 November 2026 · doi.org/10.5281/zenodo.24223975 · Bogdan Răduță, Head of Research · bogdan.raduta@flowx.ai

© 2026 FlowX.AI. Text and code licensed under the Apache License, Version 2.0; the AWS CardDemo programs keep their own Apache-2.0 licence and notice. Preprint, not peer reviewed. The archived release at the DOI above also holds the paper in its plain edition; the data, code and protocol are at github.com/flowx-ai/carddemo-equivalence-study.