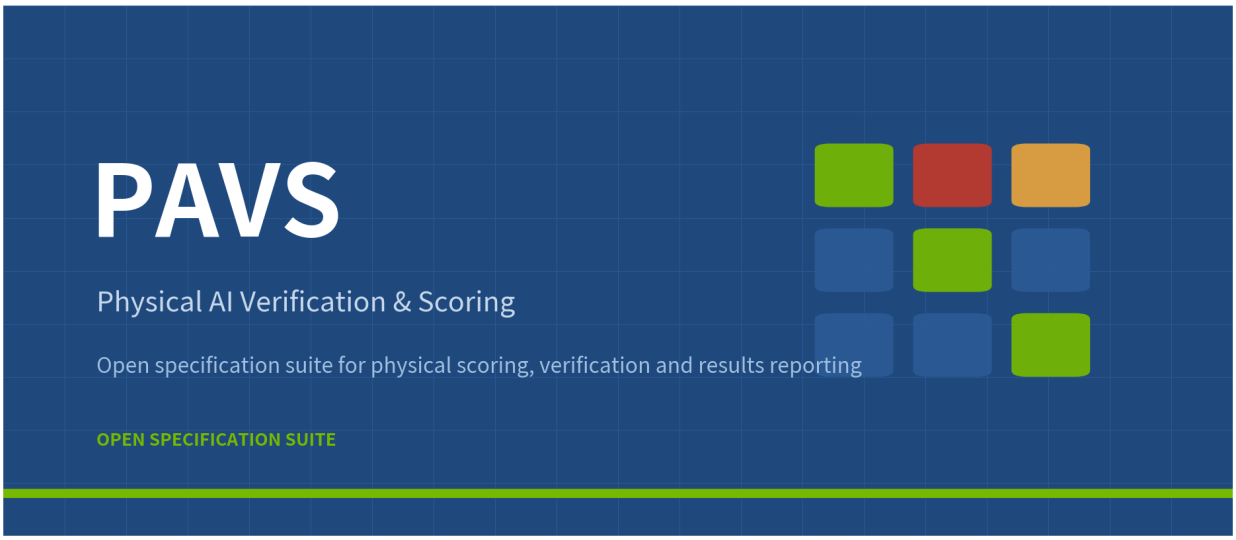


Part 1 — Specification (normative) · English translation

Issued edition v1.5 · 29 October 2026

Published by ImmersiveCast Co., Ltd. · open specification at the proposal stage (not a standard of a formal standards body)

English translation — the Korean text is authoritative · immersivecast.com/pavs

Contents

Foreword (informative)

1. Scope

2. References

3. Terms and definitions

4. Document conventions

5. Item requirements — PAVS-ITM

6. Measurement requirements — PAVS-MEA

7. Verdict requirements — PAVS-VDT

8. Scoring system verification requirements — PAVS-VRF

9. Results reporting requirements — PAVS-REP

10. Operations requirements — PAVS-OPS

11. Conformance

12. Maintenance

Annex A (normative) — Complete list of test assertions

- A.1 Items — ITM (applicable class: C1)
- A.2 Measurement — MEA (applicable class: C1)
- A.3 Verdict — VDT (applicable class: C1)
- A.4 Verification — VRF (applicable classes: C1 and C2)
- A.5 Reporting — REP (applicable class: C2)
- A.6 Operations — OPS (applicable class: C2)
- A.7 Conformance — CNF (applicable class: common)

Annex B (informative) — Clause-to-case correspondence table

Annex C (normative) — Test methods

- C.0 Common rules
- C.1 ITM — Item test methods (10)
- C.2 MEA — Measurement test methods (8)
- C.3 VDT — Verdict test methods (8)
- C.4 VRF — Verification test methods (16)
- C.5 REP — Reporting test methods (15)
- C.6 OPS — Operations test methods (10)
- C.7 CNF — Conformance test methods (5)

Annex D (informative) — Comparison table of predecessor concepts

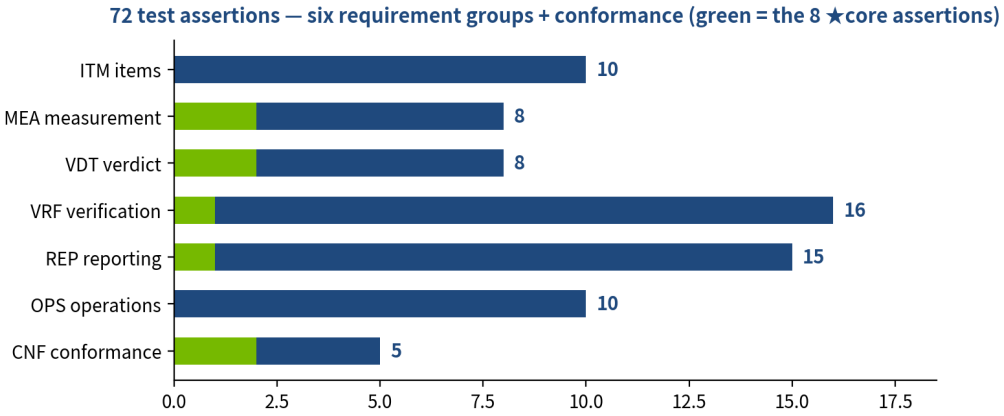
Physical AI Verification & Scoring — open specification for physical scoring, verification and results reporting

> **English edition — translation.** This is the English translation of the Korean edition v1.5 (2026-10-20). The Korean text is authoritative; where the two differ, the Korean text prevails. Assertion identifiers, clause numbers, case identifiers, versions, dates and figures are identical to the Korean edition.

> **Document status**: Issued edition v1.5 · 2026-10-29 (errata edition — no change to assertion requirements · 7 corrections to test methods · 12.6.2) · previous issue v1.4 2026-10-19 · first public release v1.2 2026-09-09 · published at immersivecast.com/pavs **Publisher**: ImmersiveCast Co., Ltd. · **open specification at the proposal stage** (not a standard of a formal standards body) · status, revision notice and revision history of this edition: 12.5–12.7
> **★Conflict-of-interest notice (12.1)**: The publisher of this specification and the author of the Solution Profile (Part 4 — unpublished, 12.5) **are the same.** That is, the party that wrote the specification and the party that is scored under it are the same. Neutrality is assured only by the product and organization neutrality of the text (the body sets no requirement favorable to any particular product), and **is not assured by the independence of the publisher.** Transfer to a third-party publisher is a goal of this specification but has not taken place in the current edition.

> **Normativity**: This document is **normative** except for the stated exceptions — NOTEs, EXAMPLEs, clauses marked "informative" and the Foreword.

Foreword (informative)



1. Purpose of the specification. This specification specifies requirements for the methods of scoring Physical AI systems in a simulation environment, verifying that scoring system and reporting the results. In the evaluation of the capability and safety of Physical AI, the layer of "how to measure in simulation, how to trust that measurement and how to report the results" is empty between "what to measure" (existing robot standards) and "the obligation to evaluate" (new AI and machinery regimes), and the purpose of this specification lies in defining that layer.

2. Summary of main contents. This specification consists of 6 requirement groups — items (Clause 5), measurement (Clause 6), verdict (Clause 7), verification (Clause 8), reporting (Clause 9) and operations (Clause 10) — and conformance (Clause 11), and specifies 72 test assertions (Annex A) and **test methods for every one of them (Annex C)**. Each clause has been generalized from the records of incidents experienced in the actual operation of simulation-based robot verification — silent false passes, silent false fails and

the self-deception of measuring instruments. The correspondence between clauses and incidents is given in Annex B and in the Part 2 Casebook (commentary).

> NOTE — An item is a unit check that produces a verdict (3.3), and a test assertion is an independent statement setting out one testable condition (3.4). Where the screens and results documents of an implementation call the same concepts by everyday words, the correspondence follows the NOTE to Clause 3.

3. Relationship to implementations. The body of this specification is neutral with respect to particular products, organizations and simulation engines and contains no trademarks. Examples showing that the requirements are executable are included anonymously in Part 3 (Reference Implementation & Reproduction — published). Part 4 (Solution Profile), which names one implementation and describes the structure by which it meets each requirement, is unpublished and is the next document scheduled for publication (12.5). Part 3 and Part 4 are informative documents, and this specification does not warrant the conformance of any particular implementation. The extent to which a particular implementation meets the requirements is stated in the Conformance Statement (11.3) of that implementation.

4. Comparison with cited standards. No standard that specifies, as normative requirements, the **layer** addressed by this specification — the measurement authenticity, verdict integrity and score reporting of scoring produced in physical simulation — was found within the scope of the survey (based on publicly available material as of 2026-08). **However, adjacent normative documents do exist, and this specification inherits from them.**

- **Methodology of conformance testing**: The ISO/IEC 9646 series established the 3-valued verdict (pass/fail/inconclusive) and the Implementation Conformance Statement (ICS). Clause 7 and 11.3 of this specification inherit them; what this specification adds is the arithmetic isolation that ‘not-measured is folded into neither the denominator nor the numerator of either the pass rate or the fail rate’ (Annex D). The single exception is the absence of measurement of a mandatory item — an absence of measurement of a mandatory item may be counted as a fail for the reason ‘absence of measurement’, and even in that case the count of not-measured results is retained separately (**PAVS-VDT-01**).
- **Testing of AI systems**: ISO/IEC TS 42119-2:2025 tailored ISO/IEC/IEEE 29119 to AI on a risk basis (published 2025-11). That layer is the **test process**; the layer of this specification lies beneath it — the provenance of physical-quantity measurement and the establishment of verdicts. It is not a competing layer but a lower one.
- **Reliability of test results**: ISO/IEC 17025 specifies the competence, uncertainty and decision rules of testing laboratories. Clauses 9 and 10 of this specification (range operator) carry those requirements into the context of simulation scoring and do not replace accreditation schemes.
- **Model credibility**: ASME V&V 10/20/40 and NASA-STD-7009 require credibility of computational models in proportion to decision influence. This specification sets no credibility ‘grade’ and specifies only the honesty of scoring; the relationship is therefore complementary.

In an adjacent domain (automated driving), scenario-based evaluation and safety standards exist (the ISO 34501–34504 ‘Test scenarios for automated driving systems’ series (2022–2024), UL 4600, the safety evaluation standard for autonomous products (first edition 2020), and others), but they do not address the simulation scoring, verification and reporting of robots and Physical AI. Existing robot standards (ISO 9283, ISO 10218, the ISO 18646 series, ISO 3691-4, etc.) specify measured characteristics and physical test methods, and this specification addresses the layer at which those measured characteristics are reproduced, measured, scored and reported in simulation — a relationship of extension and

complementation, not replacement. This specification does not imply conformance certification to those standards.

1. Scope

This specification applies to **systems for scoring, in a simulation environment,** Physical AI systems — systems equipped with artificial intelligence that perform perception, locomotion and manipulation in the physical world, and the policies and models that constitute them — and specifies the following.

- Requirements for the description and verification of test scenarios and scoring items (Clause 5)
- Trustworthiness requirements for the measurement used in scoring (Clause 6)
- Rules for producing verdicts (Clause 7)
- Requirements for the verification of the scoring system itself (Clause 8)
- Requirements for results reporting and score reports (Clause 9)
- Procedural requirements for the organization operating verification (Clause 10)
- Method of determining conformance (Clause 11)

The following are outside the scope of this specification: pass thresholds of individual domains (matters for reference standards and stakeholder agreement), procedures for conducting tests on physical hardware (real robots), statutory certification and accreditation schemes, and the implementation of any particular simulation engine.

> NOTE 1 — The Scope contains no requirements (a convention of ISO/IEC Directives Part 2).

> NOTE 2 — That this specification does not specify pass thresholds follows the same structure as a general-requirements standard for testing bodies not specifying pass criteria for individual test items. Thresholds are the province of reference standards and stakeholder agreement, and this specification specifies whether those thresholds are honestly measured, judged and reported — the source of a threshold is exposed in the score report through **PAVS-REP-02** (attribution disclosure).

2. References

Normative references: This specification has no normative references indispensable for its application — this specification defines conformance in a self-contained manner.

> NOTE — A revised edition that introduces normative references applies the following standard wording: "The following normative references are indispensable, in whole or in part, for the application of this specification. For dated references, only the edition cited applies. For undated references, the latest edition (including all amendments) applies."

Informative references

Anchoring of measured characteristics — ISO 9283 (manipulator performance), ISO 10218:2025 (industrial robot safety), the ISO 18646 series (service robot performance), ISO 3691-4 (driverless industrial trucks).

Lineage of conformance testing methodology — the **ISO/IEC 9646 series** (OSI conformance testing methodology: 3-valued verdict pass/fail/**inconclusive**, Implementation Conformance Statement **ICS**/**PICS**), W3C QA Framework (structure of conformance classes), IETF BCP 14 (normative keywords), ISO/IEC Directives Part 2 (drafting rules).

Reliability and reporting of test results — **ISO/IEC 17025** (general requirements for the competence of testing laboratories: measurement uncertainty, assurance of the validity of results, decision rules for statements of conformity), **W3C PROV** (provenance data model).

AI and software testing — the **ISO/IEC/IEEE 29119 series** (software testing), **ISO/IEC TS 42119-2:2025** (overview of AI system testing — 29119 tailored to AI on a risk basis, published 2025-11), subsequent parts of the ISO/IEC 42119 series (under development).

Model credibility and simulation verification — **ASME V&V 10 / V&V 20 / V&V 40** (verification and validation of computational models, credibility requirements in proportion to decision influence), **NASA-STD-7009** (credibility assessment of models and simulations), MIL-STD-3022 (VV&A documentation format), UNECE R157 and NATM (precedents for institutionalizing simulation verification).

> NOTE 1 — The **3-valued verdict** (Clause 7) and the **Implementation Conformance Statement ICS** (11.3) of this specification inherit concepts established by the ISO/IEC 9646 family and are not inventions. A comparison of what was inherited with what this specification adds is given in **Annex D (comparison table of predecessor concepts, informative)**. NOTE 2 — These are objects of anchoring and alignment, and this specification does not imply conformance certification to them. NOTE 3 — The standard numbers, versions and publication dates in this clause are based on publicly available material as of 2026-08. This clause contains no clause-level citations.

3. Terms and definitions

For the purposes of this specification, the following terms and definitions apply.

3.1 SUT (system under test) — A Physical AI system, robot, policy or combination thereof that is the object of scoring.

3.2 scenario — An executable document describing, in a domain language, the initial conditions, environment, tasks and events of a test.

3.3 item — A unit check that produces a verdict on the result of a scenario run. It consists of one or more test assertions.

3.4 test assertion — An independent, self-contained statement of a measurable and testable condition. It has a unique identifier.

3.5 measurement provenance — Machine-readable proof of the path (measured/commanded value/default value/estimate) by which a value used in scoring was produced.

3.6 measured — A value produced by the actual operation, during test execution, of a measurement channel that senses the physical quantity concerned. Commanded values, configured values and default values are not measured.

3.7 missing — The state in which no measurement sample exists. Missing is distinct from the value 0.

3.8 verdict — The 3-valued result of pass/fail/**not-measured** for an item or for a test as a whole, together with its grounds.

3.9 not-measured — A third state in which no verdict can be given because the measurement or stimulus required by the item is not established. It is neither a pass nor a fail. However, an absence of measurement (3.29) of a mandatory item (3.28) may be counted as a fail for the reason ‘absence of measurement’, and even in that case the count of not-measured results is retained separately, distinguished from fails for absence of measurement (**PAVS-VDT-01**).

3.10 negative control — A scenario designed to deliberately violate a specific item. It includes the reverse-direction control, in which only the stimulus of the original scenario is changed to the opposite direction. It is used to verify the discriminating power of the scorer.

3.11 null control — In an improvement claim, a control condition that is passed only through the same processing path, excluding the claimed treatment effect.

3.12 golden set — A set of test results whose verdicts have been settled and frozen. The reference for verifying verdict preservation when the scorer is changed.

3.13 effective value — The value actually applied by the system, not the value configured or submitted.

3.14 injection test — A test in which the tester deliberately produces a condition that an assertion prohibits or requires to be detected, and observes whether the test target detects, blocks or records it. It confirms not the ‘existence’ of a checking mechanism but its ‘operation’.

3.15 placebo control — In an injection test, a condition in which nothing is injected or only a harmless change is made. It determines whether a checking mechanism reacts to ‘the fact of change itself’ rather than to the defect, and whether it misfires under conditions in which it shall not fire.

3.16 evidence level — A grade of the evidential strength of a verification result: L0 (implementation exists), L1 (unit test), L2 (end-to-end execution on the production path) and L3 (direct inspection of outputs).

3.17 boundary note — An item-level statement of the scope of meaning of a pass verdict (runtime guarantee/capability boundary/absence of stimulus, etc.).

3.18 dead-end registry — The official register recording failed verification and measurement approaches, the evidence refuting them, and the preconditions for a retry.

3.19 manifest — A machine-readable document declaring the technical specifications of the SUT (geometry, joint limits, actuation limits, etc.).

3.20 reference implementation — A public reference system that implements all test assertions of this specification in executable form.

> NOTE — This definition sets the target scope of the reference implementation. The range of test assertions that the current reference implementation implements in executable form, and the extent to which they are met, are disclosed in the Conformance Statement and in Part 3.

3.21 profile — A set of items selected for each domain (for example, manipulation, mobility and safety).

3.22 range operator — An organization that conducts tests and issues results in accordance with this specification.

3.23 score report — A document, issued by a range operator, that describes the test results of the SUT (item-level verdicts and their grounds). It is not a statutory test report or certificate.

3.24 results document — Any document generated from the verdict of record that describes test results or the results of the scoring system. It includes the score report.

> NOTE — The score report is one form of results document. This specification uses no other name with the same meaning (such as ‘score document’) besides these two terms.

3.25 verdict of record — The machine-readable original record of the item-level verdicts produced by the scoring system, together with their grounds and aggregates. Results documents are generated from this record (**PAVS-REP-13**).

3.26 scoring engine — The component of a scoring system that takes measurement records as input and produces item-level verdicts and the aggregate for the test as a whole.

3.27 verdict surface — An entry point or surface of a scoring system that produces or displays a verdict or a value equivalent to a verdict (pass, fail, not-measured, ‘normal’, full score, etc.). It includes item evaluation entry points, measurement channels, and issuing surfaces such as score report, aggregation and display layers.

3.28 mandatory item — An item declared in advance as one for which a pass cannot be established without its measurement.

3.29 absence of measurement — The state in which the measurement required by a mandatory item is not established. Its counting follows **PAVS-VDT-01**.

3.30 self-declaration — The value of a verdict, of success or failure, or of an execution attribute that a scenario or SUT states about itself without a measurement basis (for example, a pass or fail declared by a scenario, or a ‘physics path’ label in an execution record). It is not measured.

3.31 positive control — A control condition known by its construction to produce the expected result (a pass verdict, or detection and return of the entries that are required to be detected and returned) if the test target is sound. Paired with a negative control (3.10), it distinguishes a scorer that fails every input from a scorer that actually catches violations.

3.32 no-stimulus control — In a measurement channel test, a control condition executed along the same path without injecting the reference stimulus. It determines whether a marker value appears without stimulus.

3.33 Abbreviations. PAVS (Physical AI Verification & Scoring), SUT (System Under Test), ICS (Implementation Conformance Statement) and ITM/MEA/VDT/VRF/REP/OPS/CNF (assertion group symbols — see 4.2).

> NOTE — The screens and results documents of an implementation can use everyday words instead of defined terms. EXAMPLE: item (3.3) → ‘specification item’; 1 run of a test forming a sample → ‘trial’. Because the same concept then has two names, recording that correspondence in the results document as well lets readers move back and forth between the specification and the screens. Where ‘test item’ on a screen denotes one test according to a scenario (3.2), it is a concept different from an item.

4. Document conventions

4.1 Normative vocabulary. The requirements of this specification are expressed by the following endings: requirement **(shall)** = ‘~하여야 한다’, prohibition **(shall not)** = ‘~하여서는 안 된다’, recommendation **(should)** = ‘~하는 것이 좋다’, permission **(may)** = ‘~해도 된다’, possibility and capability **(can)** = ‘~할 수 있다’. ‘~할 수 있다’ is a statement of possibility and is neither a permission nor a requirement. “must” is used only when referring to legal constraints external to the document. The negative permissive form ("may not") is not used.

4.2 Clause identifiers. Each requirement and recommendation is assigned a test assertion identifier of the form `PAVS-<group>-<number>`. Groups: ITM (items), MEA (measurement), VDT (verdict), VRF (verification), REP (reporting), OPS (operations) and CNF (conformance). Identifiers are not reused in revised editions (on withdrawal, the number is kept as a withdrawn number).

4.3 Executability. Every shall clause shall be objectively verifiable, and is mapped, in the assertion table of Annex A, to a checking method (automated check/document audit/demonstration by execution).

4.4 NOTEs and EXAMPLEs are informative and contain no requirements.

5. Item requirements — PAVS-ITM

The requirements of this clause shall be met by scorer implementations (Class C1).

PAVS-ITM-01 (shall) Test scenarios shall be described as documents in a self-contained domain language (a declarative language that describes scenarios so that humans can read them and machines can execute them). A third party shall be able to reproduce the same test from the same description alone, and scenario logic and data shall not be dispersed into external auxiliary code or embedded in the code of the scoring system.

> NOTE — The criterion for judging reproduction is the production of the same verdict and the same measurement coverage.

PAVS-ITM-02 (shall) Each item shall consist of one or more test assertions, and each assertion shall correspond 1:1 to an executable check. A requirement that cannot be executed cannot be an item.

PAVS-ITM-03 (shall) Each item shall declare the physical quantity it requires (including the unit), and the scoring system shall not score an item with a physical quantity different from the unit required by that item. In the event of a unit mismatch, the verdict shall be treated as not-measured and the reason shall be recorded.

> NOTE — A clause number is not a subject. Because the same clause number in a reference standard can denote different physical quantities depending on the edition, anchoring is done by the declaration of the physical quantity, not by the number. (Originating incident: a velocity of 0.8 passed a force item of 5.0 as "0.8<5.0" — see the Part 2 Casebook.)

PAVS-ITM-04 (shall) It shall be assured that a metric used for verdicts and monitoring is measured in every kind of test to which it applies. The distinction between missing and a measured 0 follows **PAVS-MEA-02**.

PAVS-ITM-05 (shall) The discriminating power of an item shall be classified from measured results, and the classification shall be recomputed whenever the test history is updated (at least every issue cycle): discriminating / constant pass / constant fail / not qualified. An item for which only a single verdict has been observed (constant pass or constant fail) shall be indicated as being in the boundary-unverified state.

PAVS-ITM-06 (shall) Boundary verification of an item for which only a single verdict has been observed shall be performed only with a reverse-direction control — a scenario in which only the stimulus of the original scenario has been changed to the opposite direction. Discrimination shall not be created by adjusting the threshold of the item.

PAVS-ITM-07 (shall) The form in which a standard item set is distributed shall be not a list of documents but an executable corpus — a set of scenarios + controls + expected verdicts.

PAVS-ITM-08 (shall) The condition axes to which an item is sensitive (density of surrounding third parties, environmental conditions, load, etc.) shall be identified and declared, and performance and safety baselines shall be defined together with the values of those conditions.

> NOTE — A baseline without conditions allows the same SUT to come out as either a pass or a fail on the same task.

PAVS-ITM-09 (should) An item set should include systematic variation along perturbation axes (object placement, camera viewpoint, initial state, instruction variants, etc.).

> NOTE — A success rate under a single condition is not evidence of generalization capability.

PAVS-ITM-10 (shall) For each rated capability published by the manufacturer (payload, speed, continuous operation, etc.), an item set shall include an item that confirms whether that capability is **sustainable under the declared conditions**. A single success shall not be taken as grounds for meeting a rated capability. Where the limit of sustainability arises from factors that the simulation does not reproduce, such as heat, current and wear, it shall be stated that the item can be met only by a test on physical hardware.

> NOTE — A rating means not a capability for a single performance but a capability for sustained work under the declared conditions. EXAMPLE: An event has occurred in which, during continuous operation while holding a load lighter than the published rated payload, the actuator of one axis reached its protective-stop threshold. That limiting factor (heat) was absent from the simulation model, and single grasp successes alone did not reveal that limit (Annex B).

6. Measurement requirements — PAVS-MEA

The requirements of this clause shall be met by scorer implementations (Class C1).

PAVS-MEA-01 (shall) Every value used in scoring shall be accompanied, in machine-readable form, by its measurement provenance (measured/commanded value/default value/estimate). An item that requires measured values shall be scored only with values whose provenance is measured.

PAVS-MEA-02 (shall) At the measurement record layer, missing data shall not be replaced with 0 or a default value. A value for which no measurement sample exists shall be recorded as "none", and a conversion that disguises absence as a value shall not be put in place.

PAVS-MEA-03 (shall) The liveness of measurement inputs used for verdicts and corrections shall be verified: if, under a pre-declared stimulus condition and observation window, the input state changes while the measurement value remains unchanged, this shall be treated not as a valid signal but as a measurement fault.

PAVS-MEA-04 (shall) Confirmation of the soundness of a measurement channel shall be performed as an independent check that injects a known reference stimulus and confirms it at the end-point output. Reading back configured values internal to the channel (a self-consistency cross-check) is not accepted as evidence of soundness.

PAVS-MEA-05 (shall) A proxy metric that is a function not of a physical quantity but of numerical-analysis conditions (integration step, etc.) shall state its qualification — the range and conditions within which it is comparable — and shall not be used for comparison of absolute values between results obtained under different conditions or for direct comparison with real-environment values.

PAVS-MEA-06 (shall) Where a measurement channel is newly established or changed, it shall be confirmed that valid values (values that are not missing, all 0, all null or a constant series) are recorded in the output data of at least 1 actual test. All-0, all-null or constant series produced without error shall be treated not as normal but as candidate measurement defects.

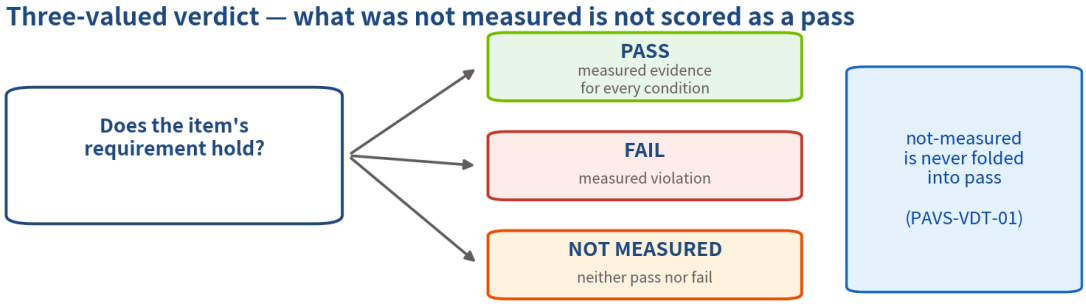
PAVS-MEA-07 (shall) For every test condition used in scoring for which a means of confirming the effective value has been declared, cases in which the submitted or configured value is not actually applied shall be detected, and the test results shall report the requested value, the effective value and whether it was applied, stated side by side.

> NOTE — Confirmation of effective values in comparative tests is specialized by **PAVS-VRF-03**.

PAVS-MEA-08 (shall) The scoring system shall not embed performance limits of the SUT (speed, torque, etc.) as constants in the system code. All restrictions shall derive from the

manifest of the SUT, and where test conditions are enforced differently from the design conditions of the SUT, that fact shall be stated in the results.

7. Verdict requirements — PAVS-VDT



The requirements of this clause shall be met by scorer implementations (Class C1).

PAVS-VDT-01 (shall) A verdict shall be 3-valued: pass/fail/not-measured. Not-measured shall not be folded into the denominator or the numerator of either the pass rate or the fail rate. However, an absence of measurement of a **mandatory item** (an item declared as one for which a pass cannot be established without its measurement) may be counted as a fail — in that case it shall be possible to identify from the record that the reason for the fail is 'absence of measurement', and the not-measured count shall be preserved and indicated as a separate state distinguished from fails for absence of measurement.

> NOTE — Counting an absence of measurement of a mandatory item as a fail is the strictest form of 'not folding not-measured into a pass' (in the same direction as the fail-loud of **PAVS-VDT-04**), and forcing it to be left only as not-measured results in a lenient overall verdict for a run that omitted a mandatory measurement (for a case, see the Part 2 Casebook). What is permitted is 'counting as a fail', not 'erasure of the not-measured state'.

PAVS-VDT-02 (shall) A pass shall be produced only on the positive evidence that every required condition is met by measurement. The default-value path and the unmet-condition path of the verdict logic shall not return a pass.

PAVS-VDT-03 (shall) Success metrics shall be cross-validated against independent evidence of the physical events that they presuppose (change of state of the object, video, contact measurement, etc.). The computation of a metric shall be conditioned on whether effective operation occurred, so that a no-op of the execution subsystem is not counted as a success.

PAVS-VDT-04 (shall) At the verdict layer, a measurement failure shall not be compensated for with substitute values, default values or mock values but shall be explicitly reported as a failure (fail-loud — the prohibition of replacement at the measurement record layer is **PAVS-MEA-02**). Mock data shall not be used outside isolated unit tests, and the fact and scope of its use shall be indicated in the results.

PAVS-VDT-05 (shall) In a test that professes physics-based scoring, results produced by bypassing a defect of the physics simulation path through a non-physical (kinematic) path shall not be used as scoring evidence.

PAVS-VDT-06 (shall) The range operator shall declare in a document the verdict layers and their strictness order, and a more lenient layer shall not overwrite with a pass a fail or warning produced by a stricter layer in the declared order.

PAVS-VDT-07 (shall) Where a defect in the verdict criteria has been corrected, all existing verdicts produced during the defect period shall be retrospectively audited against the new criteria, the results (direction and magnitude of verdict shifts) shall be left in the operational record, and the recipients of the affected score reports shall be notified.

> NOTE — Correction of a defect in the verdict criteria constitutes a ‘change’ under **PAVS-VRF-02**, and re-scoring of the golden set can be performed as a subset of this retrospective audit.

PAVS-VDT-08 (shall) For a test directive that cannot be performed (a pose or condition outside the SUT manifest), the test itself shall be rejected rather than silently approximated and scored, and the reason for the rejection shall be reported.

> NOTE — Criteria for the disposition of ineligibility: infeasibility that can be determined before test intake (manifest violation) is rejected (this clause); item-level ineligibility after execution (unit mismatch, absence of stimulus) is treated as not-measured (**PAVS-ITM-03**, 3.9).

8. Scoring system verification requirements — PAVS-VRF

The requirements of this clause shall be met by scorer implementations (C1) and range operators (C2).

PAVS-VRF-01 (shall) The range operator shall permanently maintain negative control scenarios in the regression suite. Each negative control shall declare its expected verdict (fail, with the item concerned specified), and the suite shall automatically compare the actual verdict with the expected verdict for agreement.

PAVS-VRF-02 (shall) When the scoring system is changed, it shall be confirmed that the re-scoring results of the golden set, whose verdicts have been settled, agree with the existing verdicts. Where verdicts have shifted, the number and direction of the shifts shall be counted and disclosed in the operational record, and their justification shall be recorded.

PAVS-VRF-03 (shall) The independent variable of a comparative test (A/B) shall be confirmed individually in each test group by its effective value, not by the configuration intent. The result of a comparative test whose effective values have not been confirmed shall be classified not as a refutation but as unverified.

PAVS-VRF-04 (shall) Before a test failure is attributed to a defect of the SUT, the soundness of the measurement chain itself shall first be confirmed by a refutation test.

PAVS-VRF-05 (shall) The simulation model of the SUT (geometry, joint limits, actuation limits and material properties) shall be verified against the manufacturer's confirmed specifications, and verdicts produced with an unverified model shall indicate that fact. **Results produced with a model for which a mismatch has been found in the comparison shall not be used as grounds for a conformance claim** (strengthened in v1.1 — an indication alone does not prevent those results from being cited). Where the specification of record exists in multiple locations, the location that the physics engine actually enforces shall be identified and synchronized. **Where the manufacturer publishes multiple values for the same entry, the value adopted as the baseline and the reason for its adoption shall be declared.** For entries in which the model value differs from the baseline, **the grounds for the difference shall be recorded for each entry, and a difference for which no grounds are recorded is treated as unverified.**

> NOTE 1 — What this requirement checks is **not the agreement of values but the explainability of differences**. The manufacturer itself sometimes gives two values for the same entry — there is a case in which a hardware limit and a description-document value reduced for the convenience of motion planning were both found in the published material. A check that looks only at the agreement of values judges **a difference whose grounds are**

> **recorded to be a defect, and a difference without grounds to be ‘merely slightly narrower’**. Both verdicts are wrong.

> NOTE 2 — Because the directions of a difference differ in risk, a report that lumps the directions together masks the risk. If the model is **wider** than the manufacturer's specification, it can pass poses that the physical hardware cannot perform (false pass); if it is **narrower**, it under-reports the capability of the SUT (false fail). The record also retains the direction.

PAVS-VRF-06 (shall) A claim of simulation–real-environment consistency shall be accompanied by the measurement method, sample and uncertainty, and a calibration value shall be adopted as authoritative only where it has passed a pre-specified held-out test.

> NOTE — Consistency is an object of measurement, not of claims.

PAVS-VRF-07 (should) Simulation calibration should give priority to identifying omitted physics rather than increasing the number of parameters.

PAVS-VRF-08 (shall) Determination of the cause of a defect shall be based on measurement data. If 2 corrections based on the same hypothesis have failed, correction shall be stopped and the matter shall be transferred to a pre-designated independent review body.

PAVS-VRF-09 (shall) A defect that reproduces deterministically (occurring in every run and in a standalone run) shall be treated first as a system defect, not as a matter of environmental or concurrency factors. Attribution to environmental factors is permitted only after comparison with normal cases and isolation by standalone reproduction.

PAVS-VRF-10 (shall) A regression (performance degradation) shall not be declared on the basis of a single observed failure alone. A verdict of regression shall be given only after the failure has been distinguished from intermittent failures by repeated tests in the same configuration.

PAVS-VRF-11 (shall) A change to scoring or verdict code shall be accompanied by an impact analysis of the usage sites of shared state and interfaces and by regression tests of a list of representative kinds of test declared in a document (at least 1 per operational profile), and the validity of existing verdicts produced in reliance on the changed code shall be re-evaluated.

PAVS-VRF-12 (shall) Claims of support scope and verification capability shall be based not on the existence of components (registration, code, configuration or successful initialization) but on evidence of passing end-to-end tests. Unverified entries shall be indicated in a state distinguished from verified entries.

PAVS-VRF-13 (shall) Safety-related guards and stop mechanisms shall be verified not merely as to "whether they are invoked" but as far as "whether they actually operate when invoked": firing shall be confirmed by an injection test with a known stimulus, and misfiring and insensitivity shall be distinguished by a placebo control.

PAVS-VRF-14 (shall) Every verdict shall be reproducible with a single command, and the seed, configuration and scoring code version shall be fixed and recorded in the results.

> NOTE — The criterion for successful reproduction is the production of the same verdict (numerical tolerances are declared by the item).

PAVS-VRF-15 (shall) If 2 or more parameter adjustments within the same structure have failed, it shall be re-examined whether the design assumptions (preconditions of application) of the component concerned hold under the current test conditions.

> NOTE — **PAVS-VRF-15** addresses the re-examination of the ‘structure’ of a defect, and **PAVS-VRF-08** the transfer of the ‘body’ responsible for determining it — the two requirements can be triggered together, complementarily.

8.16 Standing execution of self-checks. **PAVS-VRF-16** **(shall)** The self-checking mechanisms of a scoring system — negative controls, injection probes, consistency audits, tautology audits, etc. — shall be executed automatically at a defined interval or at the time of a change to the scoring system. The result of a check that has not been executed shall not be reported as ‘no violations’. Where the scope of automatic execution does not include the entire verdict surface protected by the checking mechanism concerned, the excluded scope shall be stated.

> NOTE 1 — This requirement distinguishes the ‘existence’ of a checking mechanism from its ‘operation’. The fact that a checker is implemented is not evidence that the check has been performed, and ‘0 violations’ from a check that has not been executed is not evidence of conformance but the result of non-execution.

> NOTE 2 — Whereas **PAVS-VRF-01** requires the permanent maintenance of negative controls, this clause requires assurance of the execution of self-checking mechanisms in general.

8.17 Withdrawn number. `**PAVS-VRF-17**` was proposed in the v1.1 draft as ‘declaration consistency of the SUT model’ but **was withdrawn** — the revision review confirmed that the requirement is already contained in `**PAVS-VRF-05**` (model authenticity). The event on which the proposal was based is not the absence of a clause but **VRF-05 not being met**, and that event is recorded in the VRF-05 row of Annex B. In accordance with 4.2, the identifier is not reused.

9. Results reporting requirements — PAVS-REP

The requirements of this clause shall be met by range operators (Class C2).

PAVS-REP-01 **(shall)** Every verification claim about the functions and support scope of a scoring system shall state its evidence level (L0–L3), and a pass verdict on such a claim shall be issued only on evidence at or above end-to-end execution on the production path (L2). Lower-level evidence shall not be indicated as a pass of system verification. The verdict rules for the scores of the SUT follow Clause 7.

PAVS-REP-02 **(shall)** A score report shall indicate, item by item, the attribution of the grounds for the verdict of each item — whether they derive from a reference standard or are the issuer's own definition. A score report shall not be issued under the name of a standard that is not registered in the issuing system.

PAVS-REP-03 **(shall)** A score report shall disclose, as lists rather than as ratios, which of the items subject to scoring were actually measured and given a verdict and which were not. Not applicable (N/A) shall not be counted as met.

PAVS-REP-04 **(shall)** The self-declaration of a test scenario shall be unable to **raise** a verdict, and a verdict produced on the basis of self-declaration alone shall not carry issuance authority (score report or conformance indication) (circular reasoning). Where a self-declaration has contributed to the overall verdict (including where it has lowered it), that fact shall be indicated item by item.

> NOTE — A self-declaration **lowering** a verdict is permitted. To prevent the pattern in which a scenario declares itself a fail but the scoring system ignores this and issues a pass (for a case, see the Part 2 Casebook), self-declaration acts as the ‘ceiling’ of the verdict — where the self-declaration is a fail, the verdict cannot be a pass. What is prohibited is a verdict ‘going up’

> through self-declaration, and issuance authority arising from self-declaration alone without measurement.

PAVS-REP-05 (shall) The scope of meaning of a pass shall be disclosed item by item by means of a boundary note (distinguishing runtime guarantee/capability boundary/absence of stimulus).

PAVS-REP-14 (should) A mechanism should be provided by which, when a measured result that contradicts a boundary note occurs, that boundary note is automatically flagged.

PAVS-REP-06 (shall) Success-rate-type figures shall not be reported without the number of trials and a confidence interval (or an equivalent indication of uncertainty). An improvement claim shall not be presented without the null-control result stated alongside.

PAVS-REP-07 (shall) Known simplifications, approximations and dormant defects (defects that do not manifest within the current test scope) of the scoring model shall be disclosed with their validity range and direction of distortion (optimistic/pessimistic) stated, and before the results of scenarios outside the validity range are issued, the defect concerned shall first be resolved or a qualifying indication shall first be made.

PAVS-REP-08 (shall) A "complete" claim in results reporting shall be accompanied by a formal self-report of the deliverable fulfillment status (a concrete list of achieved/deferred/omitted), and if there are entries not met, the status shall be indicated as partially achieved instead of complete.

PAVS-REP-09 (shall) Issued score reports shall be managed in an append-only ledger so that tampering, deletion and reordering are detectable.

PAVS-REP-10 (shall) Where the issuer of a score report has been involved in the adaptation or training of the SUT, that fact and the scope of independence shall be declared in the score report as a mandatory statement. A "neutral" indication is limited to cases with no involvement in adaptation.

PAVS-REP-11 (shall) A verdict restored after the fact (backfilled) shall carry a provenance indication that distinguishes it from real-time verdicts, and shall be recorded only under integrity rules that do not infringe existing verdicts.

PAVS-REP-12 (shall) A result in which a metric registered in the list of safety anomaly metrics declared by the range operator for each profile (tip-over, fall, sinking, etc.) is not 0 shall not be reported as "normal" in the overall status field of the score report.

PAVS-REP-13 (shall) Score reports and results documents shall be generated directly from the verdict of record, and no parallel layer that recomputes or reinterprets verdicts shall be placed between verdict and reporting. A results document shall include a measurement reliability section (lists of measured items and not-measured items, and values requiring attention).

PAVS-REP-15 (shall) Score reports and results documents that report scores produced in simulation shall indicate **the stage to which the simulation model has been compared with physical hardware and the status of the evidence of that correlation**. The indication shall distinguish at least the following — ① only self-consistency confirmed ② confirmed up to cross-comparison with other computation methods or models ③ compared with physical-hardware measurement under the same conditions, with the difference within a declared range. Where there is no correlation evidence, that fact shall be indicated as ‘none’, and the indication shall not be omitted.

> NOTE 1 — This requirement differs from **PAVS-VRF-06** (the evidence requirement for consistency claims). **PAVS-VRF-06** is triggered **when the operator claims** consistency, whereas this clause is triggered **even without a claim**. A scoring system without correlation evidence can also conform to this specification, but its score reports state, in accordance with

> this clause, whether or not there are grounds for their own predictive power. NOTE 2 — This distinction corresponds to the fact that what the precedents that institutionalized simulation verification took as the ‘condition of acceptance’ was measured correlation (Part 0 White Paper §1.1). This specification does not carry that condition over into a requirement, but **enables those who read the scores to know that status**.

10. Operations requirements — PAVS-OPS

The requirements of this clause shall be met by range operators (Class C2).

PAVS-OPS-01 (shall) The range operator shall maintain a dead-end registry, and shall enforce a lookup of the registry as a procedural step before starting a new investigation or retry.

PAVS-OPS-02 (shall) Registry entries shall have a status lifecycle (valid/resolved/partially resolved/undetermined), and when a verdict is corrected or withdrawn, the original record shall not be deleted but shall be preserved with the correction record linked to it.

PAVS-OPS-03 (shall) A registry entry for a failed approach shall state the preconditions under which a retry is permitted (facts to be proven first, or prerequisite fixes).

PAVS-OPS-04 (shall) The outputs and decisions of testing and verification shall be recorded in a designated official recording system (single source of truth), and reading the relevant records before starting work and correcting stale records immediately shall be established as procedure.

PAVS-OPS-05 (shall) The party performing verification shall not arbitrarily omit the procedural sequence and confirmation points specified by the client, and major steps — such as proceeding with the verdict-settlement procedure, starting tests that exceed a pre-agreed scale threshold and declaring completion — shall proceed only after explicit confirmation by the client.

> NOTE — This confirmation concerns whether the procedure proceeds, and is not a means of changing the content of a verdict (the prohibition of verdict changes is **PAVS-VDT-06**).

PAVS-OPS-06 (shall) Automatic gates on test intake and execution shall be based on conditions that can be measured, and rejection or hold by a gate shall be notified, together with the reason, through a channel that leaves a record. A test shall not be left in an unknown state through a silent skip.

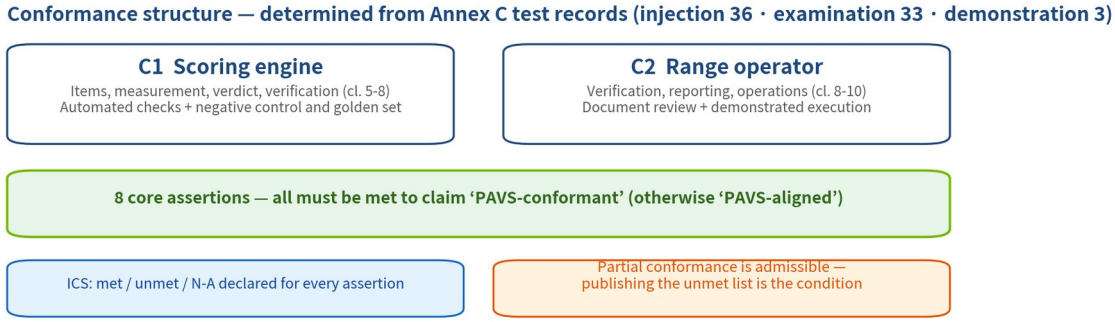
PAVS-OPS-07 (should) Where an unverified "impossible" claim about system capability is restricting the scope of functions or tests, it should be resolved by refuting or confirming it through a reproduction test.

PAVS-OPS-08 (should) The party that started a test should be responsible for it through to its normal termination and the reclamation of its resources and, instead of launching tests indiscriminately while the hypothesis is unsettled, should settle 1 test before running the next 1.

PAVS-OPS-09 (should) For the core components of the verification infrastructure, algorithms and tools proven in academia and industry should be adopted preferentially, and the infrastructure should be designed so that a new SUT can be accommodated solely by adding a declarative description (manifest), not by modifying code.

PAVS-OPS-10 (shall) The range operator shall use a scoring system that conforms to Clauses 5, 6, 7 and 8, and shall preserve the grounds for that conformance (the ICS of the scoring system used) in the operational record.

11. Conformance



11.1 Product classes. Conformance is defined for two classes.

Class	Applicable clauses	Determination method
C1. Scorer implementation	Clauses 5 · 6 · 7 · 8 in full	Automated check of the assertion table of Annex A + demonstration by execution of negative control · golden set
C2. Range operator	Clauses 8 · 9 · 10 in full	Document audit + sample audit of operating records + demonstration by execution

11.2 Conformance claim wording. **PAVS-CNF-01** **(shall)** A conformance claim shall take the following form: "This system/organization meets, for Class 〈C1|C2〉 of PAVS Part 1 v〈version〉, Annex A assertions 〈list of assertions met/all〉. Assertions not met: 〈list or none〉." A conformance claim that omits assertions not met is invalid. A conformance claim presupposes that all assertions marked '★core' in Annex A are met; a system or organization with a core assertion not met shall not claim conformance and shall use only a distinguishing expression (for example, 'PAVS-aligned').

11.2.1 List of ★core assertions. The '★core' assertions presupposed by 11.2 are the following **8**, and this list is authoritative. Where the marking in Annex A and this list disagree, **this list prevails**.

Identifier	Requirement	Reason for core status
PAVS-MEA-01	Proof of provenance · no scoring of non-measured values	If the presence of a value is read as measurement, every verdict built on it loses its basis
PAVS-MEA-02	Missing ≠ 0	If absence is replaced with 0, 'not measured' becomes 'nothing wrong'
PAVS-VDT-01	3-valued verdict (pass/fail/not-measured)	Without vocabulary to express not-measured, it is folded into one of the other two
PAVS-VDT-02	Pass = positive evidence only	If a pass without evidence cannot be prevented, the conformance claim itself does not hold
PAVS-VRF-01	Standing negative control	A check that cannot show that it is able to produce a fail is not a check
PAVS-REP-15	**Indication of the physical-hardware verification stage**	**If simulation scores can be reported while remaining**

	· status of correlation evidence	silent on their ‘predictive power for reality’, then even if all five of the above are upheld, the adopter cannot know the scope of application of those scores
PAVS-CNF-01	Form of claim wording · disclosure of assertions not met · core precondition	Without a form for claims, a claim that omits assertions not met cannot be distinguished
PAVS-CNF-02	ICS attachment · N/A eligibility	A claim without a checklist has nothing against which it can be compared

> NOTE — REP-15 was promoted to ★core in v1.0 (12.6.7). The ground for the promotion is a self-check of the reference implementation — the clause constraining the state of issuing results documents without correlation evidence lay outside the conformance preconditions, and that placement was lenient to the publisher itself. For the extent to which the implementation meets the assertions, see the Conformance Statement (public).

11.3 Implementation Conformance Statement (ICS). **PAVS-CNF-02** **(shall)** A checklist of met/not met/not applicable for every assertion in Annex A shall be attached to a conformance claim. Not applicable (N/A) may be indicated only where the class to which the assertion applies differs from the class claimed, and the reason shall be stated alongside.

11.4 Profiles. **PAVS-CNF-03** **(shall not)** A domain profile (manipulation, mobility, safety, etc.) selects a set of items, but shall not relax the requirements of this specification (Clauses 5–10).

11.4a Disclosure of profiles. **PAVS-CNF-05** **(shall)** A profile shall declare its name, item list, selection rationale and excluded items in a public document, and a claim of profile conformance shall state the profile identifier and version.

11.5 Partial conformance. **PAVS-CNF-04** **(shall)** This specification recognizes partial conformance — provided that the list of assertions not met under 11.2 is disclosed. A "conformant" indication that hides assertions not met is a violation of this specification.

12. Maintenance

> This clause specifies the procedures for **ownership, revision and objections** of the specification. An open specification without these procedures cannot be adopted, because adopters cannot confirm the point of contact for inquiries or the procedure for handling objections.

12.1 Publisher. The current publisher of this specification is stated on the cover. The publisher is responsible for **maintaining the neutrality** of the body of the specification (setting no requirement favorable to any particular product or organization), and may be the same as the author of the Solution Profile (Part 4), but shall disclose that fact on the cover.

12.2 Versions and revisions. This specification has a version in the form ‘major.minor’. **The addition or deletion of an assertion, or a change in its requirement content, increments the major version.** Editorial, explanatory and erratum corrections increment the minor version. On revision, the following shall be disclosed together — ① the list of changed assertions ② the effect of the change on existing conformance claims ③ the validity period of existing claims (transitional provisions).

> NOTE — The addition of an assertion **changes the denominator of existing conformance**

> **claims.** When the denominator changes, an implementation that was ‘conformant’ in the previous edition can cease to be conformant in the new edition, and unless the revision notice states that fact, adopters cannot know when their claims become invalid.

12.3 Objections and errata. A point of contact that receives objections and requests for errata concerning the clauses, test methods and annexes of this specification shall be made public. For each objection received, the outcome of its handling (accepted, rejected or deferred) and the reasons are kept as a public record. **Rejections are recorded too.** If the reasons for rejection are not kept, the same objection recurs, and that recurrence can be a signal of a defect in the specification. The point of contact for this edition is given in 12.5.

12.4 Revision of test methods. When a procedure of Annex C (test methods) is changed, the revision notice shall state whether existing conformance determinations made by that procedure need to be retested.

12.5 Status of this edition. This edition is an **open specification at the proposal stage** and is not a standard of a formal standards body.

Publisher = ImmersiveCast Co., Ltd. The conflict-of-interest notice required by 12.1 — the fact that the publisher is the same as the author of Part 4 — is stated on the cover (Part 4 is unpublished).

List of issued editions (the set published together with this edition; the current edition of each part other than Part 1 is determined by the document list at the place of publication — this edition does not state the edition numbers of the other parts):

Part	Edition	Status
Part 0 White Paper	Current edition = publication list	Published
Part 1 Specification	**v1.5 (this document, normative)**	Published
Part 2 Casebook	Current edition = publication list	Published
Conformance Statement	Current edition = publication list	Published
Part 3 Reference Implementation	Current edition = publication list	Published
Part 4 Solution Profile	—	Unpublished (next)

The first public release set (2026-09-09) is Part 0 v1.0, Part 1 v1.2, Part 2 v1.0 and Conformance Statement v1.0. A conformance claim cites **the version of Part 1**, and the Part 1 versions that can be cited and their deadlines are specified in 12.6.1.

Point of contact for inquiries and objections: contact@immersivecast.com (details of the objection-handling procedure to be given in a subsequent notice). The record of the handling of objections and requests for errata received follows 12.3.

★Transfer to a third-party publisher is a goal of this specification but has not taken place in the current edition. On transfer, this clause and the cover are amended and the version is incremented.

12.6 Revision notices and transitional provisions. The three elements required by 12.2 — ① the list of changed assertions ② the effect on existing conformance claims ③ the validity period of existing claims — are disclosed below for each edition. For ③, the provisions previously scattered across the individual editions have been consolidated into the single table of 12.6.1.

12.6.1 Transitional provisions (consolidated). The validity period of a conformance claim citing each edition follows the table below. Element ③ (validity period) of earlier revision notices is replaced by this table — the conflict between earlier notices (the provision that set v1.0 as the edition to be cited after the deadline for claims citing v0.6, and the provision that set v1.2 as the edition to be cited after the deadline for claims citing v1.1) has been resolved in line with the later notice.

Cited edition	Validity period	Action when moving from that edition to the next
v1.5 (current)	—	—
v1.4	**No deadline** — valid on an equal footing with v1.5 (until a revision notice of a subsequent edition provides otherwise)	When moving to citation of v1.5 — only the earlier 'conforming' determinations of the 7 assertions that had their test methods corrected (12.6.2 ②-1) are retested, limited to those assertions · all other determinations remain valid as they are
v1.3	**No deadline** — valid on an equal footing with v1.5 (until a revision notice of a subsequent edition provides otherwise)	Same as the v1.4 row — v1.4 made no change to requirements, so no retest was required (12.6.3)
v1.2	**No deadline** — valid on an equal footing with v1.5 (until a revision notice of a subsequent edition provides otherwise)	Same as the v1.4 row — v1.3 · v1.4 made no change to requirements, so no retest was required (12.6.4 · 12.6.3)
v1.1 (pre-release edition)	Until 2026-12-31	Retest of **PAVS-VDT-01** · **PAVS-REP-04** by the revised test methods (C.3.1 · C.5.4), limited to those assertions (12.6.5 ②)
v1.0 (pre-release edition)	Until 2026-12-31 (the v1.0 → v1.1 notice has no deadline provision, so the same deadline as the preceding and following editions applies)	New determination on **PAVS-ITM-10** · re-evaluation of **PAVS-VRF-05** against the strengthened wording (12.6.6 ②), followed by the action of the v1.1 row
v0.6 (pre-release edition)	Until 2026-12-31	Reflection of the promotion of **PAVS-REP-15** to ★core — if REP-15 is not met, the claim is not conformant, and REP-15 is added to the list of assertions not met (12.6.7 ②), followed by the action of the v1.0 row

Conformance claims made on or after 2027-01-01 shall cite one of v1.2, v1.3, v1.4 and v1.5. Even within the validity period, every claim shall state the version cited (**PAVS-CNF-01** form), and a claim that only says "PAVS-conformant" without a version evades the revision notice requirement of 12.2 and is therefore deemed invalid.

12.6.2 Revision notice — v1.4 → v1.5 (2026-10-20, errata edition). This edition is a correction of errata, notation and terminology and of contradictions in the wording of the test methods (Annex C), and therefore increments the minor version in accordance with 12.2. The occasion is the defects in the source text revealed by the translation comparison for the

English edition and by an independent review. The three elements required by 12.2 and whether retest is needed, as required by 12.4, are disclosed below.

① **Changed assertions: none.** There is no addition or deletion of assertions and no change in requirement content — total **72**, ★core **8**, normative keyword of each assertion (shall, should and shall not) unchanged. The word replacements in the assertion wording — ‘scoring system’ (채점 시스템) → ‘scoring system’ (채점 체계) (**PAVS-ITM-01**, ITM-03, MEA-08 and VRF-02), ‘test type’ (시험 유형) → ‘kind of test’ (시험 종류) where it refers to the kind of test subject to scoring (**PAVS-ITM-04** and VRF-11), ‘test on the real object’ (실물 시험) → ‘test on physical hardware’ (실기 시험) (**PAVS-ITM-10**), ‘placement and viewpoint’ (배치·시점) → ‘object placement and camera viewpoint’ (물체 배치·카메라 시점) (**PAVS-ITM-09**), ‘that use … as their denominator’ (분모로 쓰는) → ‘produced in reliance on’ (기대어 산출된) (**PAVS-VRF-11**) and ‘operator’ (운영자) → ‘range operator’ (검증장 운영자) (**PAVS-REP-12**) — are replacements by words of the same meaning and do not change the requirement content. The same holds for the replacement of general Korean words following the glossary of the Korean expression norm for external documents — 재산출 → 재계산 (recomputation, **PAVS-ITM-05**), 임계 → 기준값 (threshold, **PAVS-ITM-06** · OPS-05), 지령 → 명령 (command, **PAVS-VDT-08**), 등재 → 등록 (registration, **PAVS-REP-02** · REP-12) and 산출물 → 결과물 (output, **PAVS-OPS-04**); the English wording is unchanged.

② **Effect on existing conformance claims: none.** Denominator, ★core set, applicable classes and N/A eligibility unchanged. Claims citing v1.2, v1.3 or v1.4 are valid with the determinations made under the test methods of the edition cited (12.6.1).

②**-1 Re-evaluation scope of the corrections to the test methods (Annex C)** (C.0.6 and 12.4). The following 7 are corrections that align the wording of pass criteria and procedures with the clause or with other wording of the same test method, and because they can bear on the strictness of the determination criteria, they are designated **‘retest of the assertion concerned only’**. In a claim citing v1.5, the earlier ‘conforming’ determinations of these 7 assertions are indicated as ‘pending re-evaluation’ until retested (C.0.6).

- **PAVS-ITM-02** (C.1.2 procedure step 2) — the arithmetic by which the stratified sample allocation ‘at least 2 + 3’ could exceed the sample of 5 → allocation fixed at 2 and 3, with a rule for supplementing a group that falls short
- **PAVS-ITM-03** (C.1.3 pass criterion 3) — ‘not-measured cases are not removed from the denominator of the pass rate’ → included in neither the denominator nor the numerator of the pass rate (consistent with **PAVS-VDT-01** and C.3.1)
- **PAVS-ITM-06** (C.1.6 pass criterion 3) — ‘three kinds of physical basis’ → ‘one or more of the three kinds’ (consistent with procedure step 5 of the same clause)
- **PAVS-VRF-09** (C.4.9 pass criterion 3) — ‘environmental factors’ → ‘environmental or concurrency factors’ (consistent with the title and population of the same clause)
- **PAVS-REP-10** (C.5.10 pass criterion 2) — the affiliate case tested by procedure step 4 is included in the pass criterion
- **PAVS-CNF-02** (C.7.2 procedure step 1 and pass criterion 1) — ‘reference-date nonconformity’ (기준일 부적합), which was only tallied and not used, is renamed ‘reference-date defect’ (기준일 결함) and linked to the pass criterion (distinguished from the test-result word ‘nonconforming’ (부적합))
- **PAVS-CNF-04** (C.7.4 procedure step 3 and pass criterion 2) — the indication ‘assertions not met: none’ is limited to full-conformance samples, and a ‘none’ indication on a partial-conformance sample is tallied as a false indication

The other corrections to Annex C are limited to words, notation, supplying missing subjects, heading levels of clauses and unification of expressions within the same clause, and do not change pass criteria, sampling discipline or invalidity conditions; they are therefore designated **‘no retest required’**. The version of the test method applied that is entered in the test record is v1.5 in this edition (C.0.3).

③ **Transitional provisions.** Table of 12.6.1 — claims citing v1.2, v1.3 or v1.4 are valid on an equal footing with v1.5, and when moving to citation of v1.5, only the 7 assertions of ②-1 are subject to retest.

④ **List of editorial corrections.**

- Terminology: ‘scoring system’ (채점 시스템) → ‘scoring system’ (채점 체계) · the kind of test subject to scoring = ‘kind of test’ (시험 종류) (distinguished from ‘test type’ (시험 유형) of Annex C = injection · demonstration · audit) · ‘test on the real object · real-object-only’ (실물 시험 · 실물 전용) → ‘test on physical hardware · physical-hardware-only’ (실기 시험 · 실기 전용) · ‘operator’ (운영자) → ‘range operator’ (검증장 운영자) (3.22) · C.2.1 harmless copy: ‘negative control’ (음성 대조군) → ‘placebo control’ (위약 대조) (3.15) · C.2.8 ‘valid value’ (유효 값) → ‘effective value’ (실효값) (3.13) · C.3.3 ‘actuating mechanism’ (작동 기구) → ‘drive mechanism’ (구동 기구) · C.4.7 calibration ‘trial’ (시도) → ‘calibration round’ (교정 회차) (distinguished from the trial of the NOTE to 3.33) · C.4.10 ‘post-regression’ (퇴행 후) → ‘post-change’ (변경 후) · C.5.1 ‘pass’ (합격) → ‘verification pass’ (검증 합격) · C.5.9 ‘judged not-measured’ (미측정으로 판정) → ‘inconclusive’ (판정 불가) (C.0.3) · C.6.3 observability classification → ‘observable · not observable’ (관찰 가능 · 관찰 불가) · C.7.1 · C.7.2 ‘edition’ (판) → ‘version’ (버전) of Part 1 that is the subject of the claim · Part 4 name ‘Implementation Profile’ (구현 프로파일) → ‘Solution Profile’ (솔루션 프로파일) (title of the Part 4 document) · defined term 3.26 ‘scoring kernel’ (채점 커널) → ‘scoring engine’ (채점 엔진) (glossary of the Korean expression norm for external documents · same concept) · general Korean words replaced following the same glossary (whole document · same meaning · English wording unchanged): 산출물 → 결과물 (output) · 재산출 → 재계산 (recomputation) · 산출 시각 → 계산 시각 (time of computation) · 등재 → 등록 (registration; 기록 where a verdict or history is recorded) · 지령 → 명령 (command) · 임계 → 기준값 (threshold) · 갈래 → 가지 (kind) · 파지 → 집기 (grasp) · 선언값 of the manifest → 사양값 (specification value) · 공개 묶음 → 공개 자료 (public materials) · 밴드 → 허용 범위 (tolerance band)
- NOTEs: normative endings within NOTEs changed to explanatory endings (NOTE at the head of Annex C · **PAVS-VRF-05** NOTE 2 · **PAVS-REP-04** NOTE · **PAVS-REP-15** NOTE 1 · **PAVS-OPS-05** NOTE — 4.4) · ‘floor’ (바닥) → ‘ceiling’ (상한) in the **PAVS-REP-04** NOTE (the word that matches the meaning) · position of the 2 explanatory NOTEs on 8.16 (after withdrawn number 8.17 → after 8.16)
- Citations: deletion of the edition numbers of other parts (Foreword · NOTEs · Annex B · 12.5 — current edition of each part = document list at the place of publication) · Foreword 3: Part 3 ‘to be published’ → published
- Annexes: confirmation content of **PAVS-REP-03** in A (‘N/A treated as not met’ → ‘N/A not counted as met’ — same as the clause and C.5.3) · case summaries in B (B6 · D13 — Casebook titles) · **PAVS-VRF-01** row of D (‘making a paired positive control mandatory’ → ‘candidate for codification’ — current = precondition of C.4.1) · heading levels of C.0.1–C.0.6 · C.0.2 English gloss of ‘audit’ (심사) (examination → audit) · C.5.5 population attribute in the preconditions (scenario type) · C.5.14 indication of targets (consistent

with ‘three items’) · C.2.2 · C.2.4 explanation of ‘probe’ at its first occurrence · C.2.6 title · C.4.4 · C.4.16 subject of the pass criteria

- Notation: Clause 2 NOTE standard wording ‘this standard’ (이 표준) → ‘this specification’ (이 규격) · 12.1 ‘can be the same’ (동일할 수 있으나) → ‘may be the same’ (동일해도 되나) (permission ending of 4.1) · space inserted after ‘12.5’ in its heading · particle after ‘neutral’ (‘중립’로 → ‘중립’으로) · ‘not applicable’ written as one word (해당 없음 → 해당없음) · ‘all 0 · null’ → ‘all 0 · all null’ · names of the 6 common-rule clauses in the v0.5 row of 12.7 (scope added) · renumbering of revision notices (former 12.6.2–12.6.6 → 12.6.3–12.6.7)

⑤ **Candidates for review in the next formal revision (not reflected in this edition).** This carries over the list of 12.6.4 ⑤ and the 1 candidate of 12.6.3 ⑤ and adds the following.

- Unspecified elements of sampling discipline — ‘recent’ without a specified period (C.2.2, C.3.6, C.4.16, C.5.1, C.5.2, C.5.5, C.5.7, etc.); stratified sampling in which the number of strata exceeds the allocation (C.5.2, C.5.5 and C.5.6); and ‘the immediately preceding N cases’ without a reference point in time (C.6.1). Specifying these through the common rules of C.0.4 is to be reviewed. Since a change to sampling discipline cannot be designated ‘no retest required’ (C.0.6), it is handled in a formal revision.
- Consistency between test types and procedures — the procedure of C.3.8 (**PAVS-VDT-08**) is injection in its entirety (as defined in C.0.2), and C.2.4 (**PAVS-MEA-04**) and C.4.16 (**PAVS-VRF-16**) include injection steps. **PAVS-MEA-04** is added to the candidates for reassignment of test types in 12.6.4 ⑤.
- Separation of the levels of the word ‘verdict’ (판정) (SUT verdict, result of an assertion test, ICS declaration and auditor's judgment).

12.6.3 Revision notice — v1.3 → v1.4 (2026-10-19, editorial revision). This edition is a correction of terminology, explanations and notation, and therefore increments the minor version in accordance with 12.2. The three elements required by 12.2 are disclosed below.

① **Changed assertions: none.** There is no addition or deletion of assertions and no change in requirement content — total **72**, ★core **8**, normative keyword of each assertion (shall, should and shall not) unchanged. The replacement of ‘edition number’ (판번) with ‘version’ (버전) in the assertion wording is a replacement by a word of the same meaning and does not change the requirement content.

② **Effect on existing conformance claims: none.** Denominator, ★core set, applicable classes and N/A eligibility unchanged. Claims citing v1.2 or v1.3 remain valid in v1.4 with the same determinations, without retest.

②**-1 Re-evaluation scope of the corrections to the test methods (Annex C): ‘no retest required’** (C.0.6 and 12.4). The corrections to Annex C in this edition are limited to the replacement of the word ‘edition number’ (판번) with ‘version’ (버전) and to notation corrections.

③ **Transitional provisions.** Table of 12.6.1 — claims citing v1.2 or v1.3 are valid on an equal footing with v1.4.

④ **List of editorial corrections.**

- Terminology: ‘edition number’ (판번) → ‘version’ (버전) (whole document · same meaning) · NOTE to the Foreword (explanation of item and test assertion) · NOTE to Clause 3 (correspondence between defined terms and the everyday words of screens and results documents)
- NOTE to 3.20: the definition of reference implementation is the target scope; the current scope of implementation and the extent to which the assertions are met are in the Conformance Statement and Part 3

- 12.5: update of the list of issued editions (Part 0 v1.2 · Part 2 v1.2 · Conformance Statement v1.2 · Part 3 v0.5 published) · v1.4 row of the 12.6.1 transitional provisions · renumbering of revision notices (former 12.6.2–12.6.5 → 12.6.3–12.6.6)
- Annex B: correspondence for the new case D14 of the Part 2 Casebook v1.2 (VRF-09 · VRF-14 · REP-06) · indication of the Casebook version
- Notation: double angle brackets → single quotation marks · particles written without a space after Latin letters, numerals and closing parentheses (application of the domestic notation norm for external documents · meaning unchanged)

⑤ **Candidates for review in the next formal revision (not reflected in this edition).** This carries over the list of 12.6.4 ⑤ and adds the following 1 candidate.

- Determination of whether repeated runs are replicates — that results re-executed with the same seed, configuration and scoring code are identical is not an assumption but a fact to be measured (Case D14). Paired with 'deterministic repetition is not a sample (Case D11)', whether to place the criteria for determining replication (tolerance for trajectory comparison · public disclosure of verdict divergence) in the test methods of **PAVS-VRF-14** and **PAVS-REP-06** is to be reviewed.

12.6.4 Revision notice — v1.2 → v1.3 (2026-10-02, editorial revision). This edition is a correction of editorial matter, explanations and errata, and therefore increments the minor version in accordance with 12.2. The three elements required by 12.2 are disclosed below.

① **Changed assertions: none.** There is no addition or deletion of assertions and no change in requirement content — total **72**, ★core **8**, normative keyword of each assertion (shall, should and shall not) unchanged. The 3 vocabulary corrections to the assertion wording do not change the requirement content.

- `**PAVS-VDT-01**` "can be counted" (계상할 수 있다) → "may be counted" (계상해도 된다), `**PAVS-CNF-02**` "can be indicated" (표기할 수 있으며) → "may be indicated" (표기해도 되며) — aligned with the permission ending of 4.1. VDT-01 keeps exactly the intent that the v1.2 revision notice (12.6.5 ①) already described as a 'permission', and the not-applicable indication of CNF-02 keeps its earlier scope, in which it is permitted 'only where the applicable class differs'.
- `**PAVS-REP-04**` "certification indication" (인증 표기) → "conformance indication" (적합 표기) (the publisher is not a certification body) · "scenario under examination" (피검 시나리오) → "test scenario" (시험 시나리오) (unified with the defined term of 3.2).

② **Effect on existing conformance claims: none.** Denominator, ★core set, applicable classes and N/A eligibility unchanged. Claims citing v1.2 remain valid in v1.3 with the same determinations, without retest.

②**-1 Re-evaluation scope of the corrections to the test methods (Annex C): 'no retest required'** (C.0.6 and 12.4). The corrections to Annex C in this edition are limited to the following — the counts in the clause titles (C.1 '9' → '10', C.5 '14' → '15') and placement (C.1.10 and C.5.15 moved into their respective clauses) · unification of the format of C.1.10 and C.5.15 · the application wording of C.0.1 (retaining the earlier meaning that C.0 prevails where an individual test method conflicts with C.0) · correspondence of result vocabulary (NOTE to C.0.3 · Table C.0.3-1) · terminology corrections (C.5.1–C.5.3 'not satisfied' (불충족) → 'nonconforming' (부적합), C.2.4 'null control' (널대조군) → 'no-stimulus control' (무자극 대조), C.7.4 · C.7.5 'without authentication' (인증 없이) → 'without login' (로그인 없이), C.3.2 · C.5.4 'certification indication' (인증 표기) → 'conformance indication' (적합 표기) · 'scenario under examination' (피검 시나리오) → 'test scenario' (시험 시나리오), C.3.1 · C.3.3 'kernel' (커널) → 'scoring kernel' (채점 커널), vocabulary correction in the pitfall of C.4.6, C.5.15 'not-measured'

(미측정) → ‘inconclusive’ (판정 불가), deletion of incident figures from the pitfall of C.5.4) · cleanup of the ‘partially met’ (부분 충족) expression in C.7.1 (retaining the earlier scope in which an indication that declares only part of what is met is included on the not-met side) · sentence-pattern correction (pitfall of C.1.10). The strictness of thresholds, sampling discipline, invalidity conditions, procedure steps and pass criteria has not changed. The test method version of this edition is indicated as the specification version v1.3 (earlier practice).

③ **Transitional provisions.** Table of 12.6.1 — claims citing v1.2 are valid on an equal footing with v1.3.

④ **List of editorial corrections.**

- Cover: consolidation of the document status lines · relocation of the per-edition revision descriptions to 12.7 (revision history) · deletion of internal information on the document preparation process
- Foreword 3 · 12.5: list of issued editions corrected to the set actually issued (Part 3 to be published · Part 4 unpublished) · disclosure of the point of contact for inquiries and objections · correction so that the implementation fulfillment status refers to the Conformance Statement
- Foreword 4 · 3.9 · Annex D: propagation of the v1.2 revision of **PAVS-VDT-01** (absence of measurement of a mandatory item = fail for the reason ‘absence of measurement’ permitted · separate retention of the not-measured count)
- Clause 2 NOTE 3: the remaining pre-publication confirmation instruction replaced with an indication of the time of confirmation (based on publicly available material as of 2026-08)
- Clause 3: 10 new definitions (3.23–3.32 — score report · results document · verdict of record · scoring kernel · verdict surface · mandatory item · absence of measurement · self-declaration · positive control · no-stimulus control) · abbreviations 3.23 → 3.33 · ‘this standard’ (이 표준) → ‘this specification’ (이 규격) · ‘object under examination’ (피검체) → ‘SUT’ (피시험체)
- NOTEs and tables of Clauses 5–11: deletion of implementation fulfillment figures (see the Conformance Statement) · deletion of development-equipment figures (qualitative description) · abridgement of descriptions of in-house implementation incidents (see the Part 2 Casebook) · correction of clause references (**PAVS-REP-15** NOTE 1 ‘8.6’ → **PAVS-VRF-06**) · version annotations (Part 0 v1.1 · Part 2 v1.1) · conversion of interrogative statements into declarative ones
- Clause 12: consolidation of transitional provisions (12.6.1) · reordering of clauses · renumbering of revision notices (former 12.6 → 12.6.3, 12.6.0 → 12.6.4, 12.6.1 → 12.6.5) — each moved one place later in v1.4 (12.6.4 · 12.6.5 · 12.6.6) · moved one place later again in v1.5 (12.6.5 · 12.6.6 · 12.6.7)
- Annex A: ‘object under examination’ (피검체) → ‘SUT’ (피시험체) in the summary of **PAVS-VRF-04**
- Annex B: reflection of the case correspondence of the Part 2 Casebook v1.1 (re-correspondence of D6 · D10 · D11 · new cases B7 · C6 · C7 · D12 · D13 — 38 cases in Casebook v1.1) · deletion of development-equipment figures and internal information
- Annex C: the corrections of ②-1
- Annex D: **PAVS-VDT-01** exception stated alongside in the 3-valued verdict row · the statement ‘execution of checks, not document audit’ in the conformance class row corrected to the actual composition of test types in Annex C (injection 36 · audit 33 · demonstration 3) · ‘the last two rows’ in the NOTE → row names
- Footer: the count of the normative composition corrected to the tally of the normative column of Annex A (shall 66 · should 7 · shall not 2 · may 1 → shall 65 · should 6 · shall not 1 · may 0) — a correction of a tallying error; the assertions did not change

⑤ **Candidates for review in the next formal revision (not reflected in this edition).** The following are candidates that can change the requirement content and have therefore not been reflected in this edition. If adopted, they are handled as a formal revision under 12.2 (list of changed assertions, effect and transitional provisions).

- Reassignment of test types (**PAVS-VRF-16**, **PAVS-VDT-08** and **PAVS-VRF-13**)
- Assignment of applicable classes to each assertion
- Vocabulary of the conformance claim form and the eligibility for 'partial conformance'
- Codification of the positive control paired with **PAVS-VRF-01**
- Inclusion of missing data in **PAVS-REP-12**
- Test steps corresponding to the strengthened wording of **PAVS-VRF-05** (prohibition of the use of the results of a mismatched model)
- Scope of application of **PAVS-OPS-05**
- ICS handling rules for should assertions
- Definition of retest in C.0.6 and the version scheme of test methods
- 2 candidates for Annex D
- Extension of **PAVS-VRF-04** following C5 of Annex B (attribution of incidents of inability to act)
- Non-interference of measurement (Case D10)
- Deterministic repetition is not a sample (Case D11)
- Separation of compound assertions
- Consistency between the version provision of 12.2 and versioning practice (NOTE to 12.6.5)

12.6.5 Revision notice — v1.1 → v1.2 (2026-09-04). The three elements required by 12.2 are disclosed below.

① **Changes in requirement content: 2.** `**PAVS-VDT-01**` (3-valued verdict): permits counting the absence of measurement of a mandatory item as a fail for the reason 'absence of measurement', and requires the not-measured count to be retained separately. `**PAVS-REP-04**` (prohibition of self-declaration circularity): changes 'can move neither the verdict nor the aggregate' to 'cannot raise the verdict + a run with self-declaration alone has no issuance authority'. Test methods C.3.1 and C.5.4 were revised to the same effect. There is no addition or deletion of assertions — total **72** unchanged.

①**-1 Reason for the revision.** In a self-test according to the procedures of Annex C, it became apparent that the earlier wording of the two assertions departed from the intent of this specification — 'not-measured results and self-declarations are not laundered into passes'. Implemented as worded earlier, verdicts **become lenient** (conflict with C.3.4 and disregard of self-declared fails). Since the specification had wording that departed from its own intent, the wording was corrected. This is not a relaxation — the requirement of both assertions 'not to fold into a pass' is unchanged, and handling in the fail direction has merely been explicitly permitted.

② **Effect on existing conformance claims.** Denominator and ★core set unchanged. Because the determination criteria of the test methods have changed (C.0.6), the two assertions are subject to **retest limited to those assertions**. The determinations of the other 70 are not affected.

③ **Validity period of existing claims.** Consolidated into the table of 12.6.1.

> NOTE — 12.2 specifies that 'a change in requirement content increments the major version', but the practice of this specification at v0.6 (change in the requirement of VRF-05) and v1.1 (addition of ITM-10) incremented the minor version. This edition also follows that practice and is designated v1.2. The inconsistency between practice and wording is itself recorded as a subject for correction of 12.2 (or correction of the practice).

12.6.6 Revision notice — v1.0 → v1.1 (2026-09-03). The three elements required by 12.2 are disclosed below.

① **Changed assertions — 1 added · 1 strengthened · 1 proposal withdrawn.** `**PAVS-ITM-10**` (sustainability of rated capability) added — total **71 → 72**. Wording of `**PAVS-VRF-05**` **strengthened** — to the earlier "shall indicate" has been added "the results of a model for which a mismatch has been confirmed shall not be used as grounds for a conformance claim". `**PAVS-VRF-17**` (consistency of model declarations) was proposed and then **withdrawn** — the revision review confirmed that its requirement already exists in `**PAVS-VRF-05**`. The incident on which it was based (the manifest and the simulation asset for the same arm coexisting while declaring different masses) is **not the absence of a clause but a failure to meet VRF-05**. Rewriting an old failure as a new gap only increases the denominator, not the requirements. The test method for the new clause (C.1.10) was specified in Annex C at the same time.

② **Effect on existing conformance claims.** The denominator increases to 72. Existing 'met' determinations do not change, but a new determination on `**PAVS-ITM-10**` is required, and for `**PAVS-VRF-05**` the 'met' determination shall be re-evaluated against the strengthened wording (12.2). The incidents underlying the two clauses are in Annex B. For the extent to which the implementation meets the assertions, see the Conformance Statement (public).

③ **Validity period of existing claims.** Consolidated into the table of 12.6.1.

12.6.7 Revision notice — v0.6 → v1.0 (2026-09-01). The three elements required by 12.2 are disclosed below.

① **Changed assertions.** `**PAVS-REP-15**` (indication of the physical-hardware verification stage and the status of correlation evidence) — **1 assertion**. The requirement wording did not change; **the ★core marking was added**. There is no addition or deletion of assertions — the total did not change in that revision.

② **Effect on existing conformance claims.** The ★core set increases from 7 to **8**. Because 11.2 makes a conformance claim presuppose that all ★core assertions are met, **a claim that was 'conformant' under v0.6 but does not meet REP-15 is not conformant under v1.0.** A claim that discloses its list of assertions not met (11.2 form) shall add REP-15 to that list. There is no effect on the class (C1/C2) distinction or on N/A eligibility. Since the test methods of Annex C did not change, **no retest is required** under 12.4 — the test method for REP-15 (C.5.15) has not changed since it was added in v0.5. For the extent to which the implementation meets the assertions, see the Conformance Statement (public).

③ **Validity period of existing claims.** Consolidated into the table of 12.6.1 (earlier provision: claims citing v0.6 are valid until 2026-12-31, "based on PAVS Part 1 v0.6" is to be stated, and a "PAVS-conformant" claim without a version is invalid).

12.7 Revision history (informative). The gist of the changes in each edition is recorded. For details of changes that affect verdicts, the corresponding revision notice (12.6.2–12.6.7) is authoritative.

Edition	Date	Summary
v0.2	—	Annex A (complete list of test assertions) · Annex B (clause–case correspondence table) added · Korean standards format applied
v0.3	—	5 clauses added (**PAVS-REP-13** direct link to the verdict of record · REP-14 · VRF-15 · OPS-

		10 · CNF-05) · conformance supplemented (precondition of 7 ★core assertions · N/A eligibility · profile disclosure) · classes organized · compound norms separated · verifiability wording refined in 14 places · declaration of the applicable party for each clause · anonymization of annexes
v0.4	—	**PAVS-VRF-16** added (standing execution of self-checks — originating from the reference implementation's self-check revealing that 'the checker exists but is not wired into an automatic execution point') · assertions 69 → 70
v0.5	—	Normative Annex C (test methods) added — preconditions · procedures · pass criteria · records · pitfalls for all assertions, and common rules in 6 clauses (scope · types · records · sampling discipline · invalidity conditions · re-evaluation on revision) · resolution of the 'provisional criteria' wording in Annex A · 2 terms added
v0.6	—	Change in the requirement content of **PAVS-VRF-05** — check target of model authenticity changed from 'agreement of values' to 'explainability of differences' · 3 procedure steps · 2 pass criteria · 3 record items · 1 pitfall added in C.4.5 (Case D7)
v1.0	2026-09-01	**PAVS-REP-15** promoted to ★core · 11.2.1 ★core list added (12.6.7)
v1.1	2026-09-03	**PAVS-ITM-10** added (71 → 72) · **PAVS-VRF-05** strengthened · **PAVS-VRF-17** proposal withdrawn (withdrawn number, 8.17) · C.1.10 added (12.6.6)
v1.2	2026-09-04	**PAVS-VDT-01** · **PAVS-REP-04** requirement content changed · C.3.1 · C.5.4 revised (12.6.5) · first public release 2026-09-09
v1.3	2026-10-02	Editorial revision — no change to assertion requirements (12.6.4)
v1.4	2026-10-19	Editorial revision — no change to assertion requirements ·

		‘edition number’ (판번) → ‘version’ (버전) · NOTE on the correspondence of defined terms · correspondence for Case D14 · notation corrections (12.6.3)
v1.5	2026-10-20	Errata edition — no change to assertion requirements · 7 corrections to test methods (retest of the assertion concerned only) · unification of terminology · endings in NOTEs · deletion of citations of the editions of other parts (12.6.2)

Annex A (normative) — Complete list of test assertions

Mapping of the assertion IDs and checking methods of all clauses. **The detailed test method of each assertion (preconditions, procedure, pass criteria and record items) is specified by normative Annex C (test methods)** — the conformance determination is based on the records of tests performed in accordance with the procedures of Annex C. Of the three forms of checking method — **automated** (machine-executed checks such as injection and static checks), **demonstration** (demonstration by execution and its record) and **audit** (audit of documents and operational records) — the principal method is written first. In determining conformance, this table is the denominator of the ICS (11.3).

A.1 Items — ITM (applicable class: C1)

ID	Summary	Normative keyword	Checking method	What is confirmed
PAVS-ITM-01	Self-containment of scenarios	shall	automated + demonstration	0 external code dependencies of the scenario (static check) · reproduction run by a third party
PAVS-ITM-02	Item = executable assertions, 1:1	shall	automated	audit of the existence of a check mapping for every item
PAVS-ITM-03	Declaration of physical quantities (units) and rejection of mismatches	shall	automated	injection of a unit mismatch → handled as not-measured + reason recorded
PAVS-ITM-04	Guaranteed measurement of metrics in all kinds of test	shall	automated	audit of the guaranteed measurement of metrics by kind of test (the distinction of missing falls under MEA-02)

PAVS-ITM-05	Classification of discriminating power by measurement and its recomputation	shall	automated + audit	production and recomputation of the discriminating-power classification · boundary-unverified indication of single-verdict items
PAVS-ITM-06	Reverse-direction controls only · no threshold adjustment	shall	audit + automated	boundary verification history · comparison of the threshold change history
PAVS-ITM-07	Distribution as an executable corpus	shall	audit	actual existence of scenarios + controls + expected verdicts
PAVS-ITM-08	Baselines with conditions	shall	automated + audit	condition field made mandatory in baseline definitions
PAVS-ITM-09	Variation along perturbation axes	should	audit	whether the item set includes perturbation axes
PAVS-ITM-10	Sustainability of rated capabilities	shall	audit	existence of sustained-test items for each rating · single success not accepted · statement of whether physical-hardware-only

A.2 Measurement — MEA (applicable class: C1)

ID	Summary	Normative keyword	Checking method	What is confirmed
PAVS-MEA-01	★core — proof of provenance · no scoring of non-measured values	shall	automated	existence of provenance metadata for values · rejection on injection of non-measured values
PAVS-MEA-02	★core — missing≠0	shall	automated	injection of missing data → 0 substitutions (absence-pass probe)
PAVS-MEA-03	Measurement liveness	shall	automated	injection of an unchanging signal

				→ handled as a fault
PAVS-MEA-04	End-point probe with a known stimulus	shall	demonstration	reference stimulus injection · record of confirmation of the end-point output
PAVS-MEA-05	Statement of the qualification of proxy metrics	shall	audit + automated	document stating the qualification of the metric · comparison outside the range blocked
PAVS-MEA-06	Confirmation of actual output from new channels	shall	audit	record of confirmation of the first measured value · all 0/null = handled as a defect
PAVS-MEA-07	Requested value/effective value/application stated side by side	shall	automated	existence of result fields · firing of mismatch detection
PAVS-MEA-08	No hard-coding of SUT limits	shall	automated + demonstration	check that limit values derive from the manifest · effective reflection of changes

A.3 Verdict — VDT (applicable class: C1)

ID	Summary	Normative keyword	Checking method	What is confirmed
PAVS-VDT-01	★core — 3-valued verdict (pass/fail/not-measured)	shall	automated	injection of not-measured → not counted as a pass (for mandatory items, a fail for the reason 'absence of measurement' permitted) · not-measured count preserved separately
PAVS-VDT-02	★core — pass = positive evidence only	shall	automated	injection of empty input → 0 passes
PAVS-VDT-03	Physical cross-validation of success metrics	shall	automated + demonstration	injection of a no-operation (no-op) → not counted as a success

PAVS-VDT-04	Measurement failure = fail-loud	shall	automated	injection of a measurement failure → failure reported · 0 substitute values
PAVS-VDT-05	No scoring of results that bypass physics	shall	automated	non-physical output → indication of non-acceptance of evidence
PAVS-VDT-06	Hierarchy of verdict authority	shall	automated	attempt at lenient overwriting → blocked
PAVS-VDT-07	Retrospective audit upon correction of a criteria defect	shall	audit	retrospective audit results attached to the defect correction history
PAVS-VDT-08	Rejection of unperformable directives	shall	demonstration	injection of a directive outside the manifest → rejection + reason

A.4 Verification — VRF (applicable classes: C1 and C2)

ID	Summary	Normative keyword	Checking method	What is confirmed
PAVS-VRF-01	★core — standing negative controls	shall	demonstration + automated	existence of cells with expected verdict = fail · firing of automatic comparison
PAVS-VRF-02	Verdict preservation of the golden set	shall	automated	re-scoring difference of 0, or shifts counted and disclosed
PAVS-VRF-03	Per-group confirmation of effective values in A/B	shall	audit	per-group effective values recorded in the comparison report
PAVS-VRF-04	Refutation of the measuring instrument before attribution to the SUT	shall	audit	record of measurement refutation before attribution
PAVS-VRF-05	Model authenticity comparison	shall	audit + automated	record of comparison with the manufacturer's specifications · indication of unverified models
PAVS-VRF-06	Consistency = object of	shall	audit	method, sample and uncertainty ·

	measurement · held-out adoption			history of passing the held-out test
PAVS-VRF-07	Omitted physics first	should	audit	record of approaches in the calibration history
PAVS-VRF-15	Re-examination of design assumptions after 2 failures	shall	audit	record of re-examination of assumptions after repeated failures
PAVS-VRF-16	Standing execution of self-checks · prohibition of reporting non-execution as 'no violation'	shall	automated + audit	automatic execution history · comparison of the check scope with the verdict surface · statement of the excluded scope
PAVS-VRF-17	**Withdrawn number (abolished)** — proposed in the v1.1 draft and then retracted. The requirement is in VRF-05 (8.17)	—	—	no test method (not an assertion)
PAVS-VRF-08	Measurement-based determination of causes · transfer after 2 failures	shall	audit	record of the determination procedure
PAVS-VRF-09	Deterministic reproduction = system defect first	shall	audit	record of the grounds for attribution (comparison with normal cases and isolation by standalone reproduction)
PAVS-VRF-10	No declaration of regression from a single failure	shall	audit	repeated-test basis for regression verdicts
PAVS-VRF-11	Change impact analysis · re-evaluation of existing verdicts	shall	automated + audit	regression suite execution history · re-evaluation record
PAVS-VRF-12	Existence≠capability (end-to-end evidence)	shall	audit	end-to-end evidence for the support list · distinct indication of unverified entries
PAVS-VRF-13	Verification of safety guard firing	shall	demonstration	firing in an injection test + placebo control record

PAVS-VRF-14	1-command reproduction · configuration pinned	shall	demonstration + automated	reproduction run · seed/version pinning fields

A.5 Reporting — REP (applicable class: C2)

ID	Summary	Normative keyword	Checking method	What is confirmed
PAVS-REP-01	Evidence level indication · pass = L2+	shall	automated	existence of the level field · passes below L2 blocked
PAVS-REP-02	Attribution disclosure · phantom-standard gate	shall	automated + audit	firing of rejection of issuing under unregistered standards · item-level attribution indication
PAVS-REP-03	Disclosure of the coverage list	shall	automated	generation of the not-measured list · N/A not counted as met
PAVS-REP-04	No circularity of self-declaration	shall	automated	no upgrading by self-declaration · issuance authority blocked for runs based on self-declaration alone · indication of contribution
PAVS-REP-05	Item-level disclosure of boundary notes	shall	audit	item-level boundary indication
PAVS-REP-13	Results document = directly linked to the verdict of record	shall	automated	injection of tampering into the verdict of record → document mismatch detected · existence of a measurement reliability section
PAVS-REP-14	Automatic flagging of stale boundary notes	should	automated	injection of a contradicting measured value → flagging fires
PAVS-REP-15	★core — indication of the physical-hardware verification stage and the status of correlation evidence	shall	automated	document generated from a verdict without correlation evidence → confirmation that a ‘none’ indication exists

PAVS-REP-06	Statistical discipline (n, CI and null control)	shall	audit + automated	detection of the accompanying elements in numerical reports
PAVS-REP-07	Disclosure of dormant defects and approximations	shall	audit	defect registration · indication of the validity range and direction of distortion
PAVS-REP-08	Discipline of completion declarations	shall	audit	existence of a self-report section · partial≠complete
PAVS-REP-09	Integrity of the score ledger	shall	automated	detection of injected tampering, deletion and reordering
PAVS-REP-10	Declaration of conflict of interest (involvement in adaptation)	shall	automated	rejection of issuing without mandatory fields
PAVS-REP-11	Distinct indication of backfilled verdicts	shall	automated	provenance indication · existing verdicts not infringed
PAVS-REP-12	Safety anomaly≠‘normal’	shall	automated	‘normal’ indication blocked when an anomaly metric is present

A.6 Operations — OPS (applicable class: C2)

ID	Summary	Normative keyword	Checking method	What is confirmed
PAVS-OPS-01	Dead-end registry · lookup before starting	shall	audit	the registry itself · record of the lookup procedure
PAVS-OPS-02	Registry entry lifecycle · preservation on correction	shall	audit	status indication · original record preserved + correction linked
PAVS-OPS-03	Statement of retry preconditions	shall	audit	preconditions entered for each entry
PAVS-OPS-04	Recording in a single source of truth · prior reading	shall	audit	recording location · prior-reading procedure · correction of stale records
PAVS-OPS-05	Respect for the client's procedures and confirmation	shall	audit	confirmation records of major steps

	points			
PAVS-OPS-06	Gate = measured basis · visible notification of rejection	shall	automated + audit	existence of reason notifications for rejections and holds
PAVS-OPS-07	Resolution of unverified ‘impossible’ claims	should	audit	refutation/ confirmation history
PAVS-OPS-08	Responsibility for reclaiming test resources	should	audit	termination and reclamation records
PAVS-OPS-09	Verified tools first · declarative acceptance	should	audit	adoption rationale · manifest acceptance path
PAVS-OPS-10	Use of a conformant scoring system and preservation of its basis	shall	audit	preservation of the ICS of the scoring system used

A.7 Conformance — CNF (applicable class: common)

ID	Summary	Normative keyword	Checking method	What is confirmed
PAVS-CNF-01	★core — form of the claim wording · disclosure of assertions not met and the core precondition	shall	audit	conformity of the claim text to the form · existence of the list of assertions not met
PAVS-CNF-02	★core — ICS attachment and N/A eligibility	shall	audit	attachment of the checklist of all assertions
PAVS-CNF-03	No relaxation of requirements by profiles	shall not	audit + automated	profile definitions retain the requirements of Clauses 5 to 10
PAVS-CNF-04	Partial conformance = premised on disclosure	shall	audit	partial conformance indication without concealment of assertions not met
PAVS-CNF-05	Public declaration of profiles	shall	audit	public document of the profile name, list, rationale and exclusions

Annex B (informative) — Clause-to-case correspondence table

A correspondence list of the incidents from which each clause originated. **Included cases** refers to the numbers in the Part 2 Casebook (A1 to E4), and **corpus incidents** refers to the gist of incidents that are not yet included in the Casebook but whose original records are preserved (candidates for inclusion in a revised edition). No clause has an empty correspondence — every clause of this specification derives from an actual incident or from a record of failure in the published literature.

Clause	Included cases (Part 2)	Corpus incidents (basis not yet included)
ITM-01	—	Recurring incidents in which the dispersal of scenario logic into auxiliary code made reproduction and maintenance impossible (history of rework for self-containment)
ITM-02	A2 (the betrayal of an item's name)	Incident in which measurement code existed but was not wired into issuing
ITM-03	A1 (0.8<5.0 passed)	—
ITM-04	A1 · A5 (an item that the series cannot measure) · C6 (body contact outside the code table slipped through as a pass)	A metric dedicated to a particular scenario was always 0 in other kinds of test → normal tests wrongly aborted
ITM-05	A2 · A3 · C3 · A6 (constant pass due to a runtime guarantee) · A7 (a metric that measures the scenario)	Sweeping audit of tautological items (many unconditionally passing items found)
ITM-06	A3 (reverse-direction control) · A6 (a boundary that was not closed)	—
ITM-07	A3	Incident in which tests with 0 acceptance criteria were registered as passes
ITM-08	A4 (pedestrian SLA)	A change to the cell specification overrode the training premise, and the same policy failed every time
ITM-09	D11 (repeated runs are replicates, not samples)	Published literature: collapse from 95% to below 30% across 7 perturbation axes (LIBERO-Plus 2025)
ITM-10	—	Incident in which, during continuous operation while holding a load lighter than the published rated payload, the actuator of one axis reached its protective-stop threshold (measured 2026-09-03). It was not revealed by a single grasp success, and the limiting factor (heat) was absent from the simulation model

MEA-01	B1 (a constant multiple of the commanded value) · B6 (a different meaning in each test series) · C7 (a metric without proof of measurement)	—
MEA-02	B2 (22 absence-passes)	Discovery of a structure in which missing was recorded as 0, yielding a full score for a '0-second emergency stop'
MEA-03	B3 (closed loop on a faulty measurement)	—
MEA-04	B3 (self-consistency of camera read-back) · B7	—
MEA-05	B5 (representation by 161 points)	Confirmation by A/B that the contact-force metric is a function of the integration step (switch to prohibiting comparison of absolute values)
MEA-06	D10 (a view for measurement changed the physics of the SUT)	A newly created trajectory pose channel recorded all 0 because of a 3-fold defect — discovered only when the output file was opened
MEA-07	B1 · B7 (the reflected mass-change value was correct but the effect was 0) · D12	Submitted physics conditions silently ignored on a particular path (switch to stating requested/effective side by side)
MEA-08	B4 (speed cap)	—
VDT-01	A5 · C7 (value present but no proof of measurement → fail with identified reason) · D11	History of introducing a third state that folds inability to measure into neither pass nor fail (honest statement of a measurement budget of 1%)
VDT-02	C1 (default-value pass) · B2 · E1 · C6	—
VDT-03	C2 (no-op grasp)	Contact-force discrimination of a success that was a position-control artifact (weld)
VDT-04	—	Switch to a total prohibition of the early practice of bypassing failures via alternative paths and returning them as successes
VDT-05	—	The temptation to bypass physics-mode defects non-physically — codification of the exclusion of non-physical output from scoring evidence
VDT-06	E1	Structure in which, in the same test, a lenient display layer overrode the strict verdict of record
VDT-07	C1 (89% retrospective) · C6	—
VDT-08	C4 (rejection of impossible	—

	directives)	
VRF-01	D1 (testing the scorer) · E3 (control of the metric itself)	—
VRF-02	D2 · D10 · D12	Migration of the issuing system on proof of 0 difference in golden-set re-scoring
VRF-03	D3 (identical downgrade in both groups) · B7	—
VRF-04	D4 (measurement in disguise) · C5 (attribution of an incident in which action was impossible — candidate for clause extension)	—
VRF-05	C4 · C6 · D7 (the manufacturer publishes two values) · D8 (the loader overrode the declaration) · D12 (model specification defect — gravity disabled on the arm links)	Specification of record dispersed across 3 locations — identification and synchronization of the location that the physics enforces · incident in which scoring was carried out for several months while the manifest and the simulation asset declared different masses for the same arm (2026-09-02) — there was no procedure for comparing the two declarations. After correction, position accuracy improved substantially
VRF-06	D13	Incident in which an incorrect standard-error formula made the same data ‘significant’ · incident in which, without a held-out test, incorrect calibration values nearly became authoritative three times
VRF-07	—	Incident in which the root cause of 5 consecutive failed parameter adjustments was a violation of design assumptions · calibration that generalized once the parameters were reduced and the omitted physics was added
VRF-08	—	Codification of discipline for the repeated failure of guess-based corrections without measurement (transfer after 2 failures)
VRF-09	D14 (diverging verdicts on reruns of the same pose — cause determined after ruling out the environment)	Incident in which a deterministically reproduced defect was misdiagnosed as environmental, wasting half a day
VRF-10	—	‘Regression’ declared from a single failure, after which the same configuration reproduced

		the baseline the next day (intermittent failure)
VRF-11	D2 · D6 · D9 (a fix to a shared subsystem broke other robot types) · D10 · D12	Discipline of re-evaluating the denominators of existing verdicts after scoring code changes (re-measurement of consuming cells when a band is updated)
VRF-12	—	Audit confirmed that, of many registrations, very few pass end-to-end (registration≠support)
VRF-13	D5 (the guard that was dead)	—
VRF-15	—	Incident in which the root cause of 5 consecutive failed parameter adjustments was a violation of design assumptions (the ‘structural’ aspect of the same family as the VRF-07 row)
VRF-16	D6 (the checkers existed and nobody ran them) · D13 (the preregistration gate was called after execution had finished)	Incident, found in the self-check of the reference implementation (2026-08), in which probes, detectors and recomputation tools existed in a state of having passed self-verification but were not wired to automatic execution points, and the only automatic point excluded most of what it protected through scope gating
VRF-14	D11 · D14 (the identity of rerun results is a fact to be measured)	History of establishing the discipline of 1-command reproduction, seed pinning and bit-level comparison
REP-01	—	Detection and grading of the practice of reporting lower-level evidence (implementation exists, static check passed) as ‘verification passed’
REP-02	E1 (phantom standard)	—
REP-03	B2 · E1	Switch to disclosing, as a list, the entries subject to scoring but not measured (no conversion to ratios)
REP-04	E1	Automatic indication of tests in which self-declaration becomes the overall verdict when the verdict of record has too few measured items
REP-05	—	Introduction of item-level public notice of the ‘meaning’ of a pass (runtime guarantee/absence of stimulus)
REP-06	E3 · E4 (collection rate of 100% and collapse reversal) · D11 ·	—

	D14	
REP-07	—	Incident of registering the validity range and direction of distortion (optimistic) of a simplification of actuation allocation
REP-08	—	Incident in which repeated declarations of completion for partial achievement damaged trust (introduction of the formal self-report)
REP-09	—	History of building hash-chain verification of the score-issuing ledger (detection of tampering, deletion and reordering)
REP-10	—	Self-regulation of the conflict of interest in scoring an SUT whose adaptation the scorer was involved in (restriction of the neutral indication)
REP-11	—	Incident in which, among verdicts restored after the fact, the portion with 0 independent checks was withdrawn through self-audit
REP-12	C3 (sunk yet passed)	Codification of the prohibition on reporting results with a fall anomaly as ‘normal’
REP-13	E2 (direct link from the verdict of record to the report)	Incident in which a display layer recomputed verdicts with its own formula and restored passes (linked to B2)
REP-14	—	Incident in which, the day after a boundary note was assigned, the staleness guard fired in actual operation and deleted a false sentence
REP-15	—	(Newly established 2026-08-26) The basis of the clause is not a case but **the acceptance condition of a precedent** — it derives from the fact that a precedent that institutionally accepted simulation verification made ‘proof of correlation with measurement’ its condition of acceptance
OPS-01	—	Waste from retrying recorded failed approaches → institutionalization of the registry and lookup before starting
OPS-02	D4 (preservation of a withdrawal) · D13	Practice of recording the correction of a misdiagnosis by preserving the original record + linking the follow-up

OPS-03	—	Practice of registering the conditions under which a retry is permitted (facts to be proven first)
OPS-04	—	Incident in which domain outputs were left only in personal records, causing discontinuity of work
OPS-05	—	Rework caused by unauthorized omission of the client's procedure (single-case confirmation→agreement→batch)
OPS-06	D13	Sweeping audit of many guess-based pre-gates that had rejected normal tests or left them unattended
OPS-07	—	A single line of unverified ‘impossible’ comment sealed off a feature for a long period (refuted by 2 reproductions)
OPS-08	—	Abandoned test processes damaged state management, creating phantom tests
OPS-09	—	5 consecutive failures of in-house implementations → structural resolution by adopting a verified public algorithm
OPS-10	—	Making the requirements of the operator class checkable — preservation in operational records of the basis for scoring system conformance
CNF-01	—	Invalidation of ‘conformant’ indications that conceal assertions not met — generalization of the E1 (phantom standard) incident to the conformance layer
CNF-02	—	Self-report on all assertions (ICS) — the family of incidents in which partial achievement was indicated as completion
CNF-03	—	Prevention of the relaxation of requirements through profiles
CNF-04	—	Premise of honest disclosure for partial conformance
CNF-05	—	Prevention of the abuse of professing a ‘pass’ through arbitrarily small profiles

Annex C (normative) — Test methods

This Annex specifies the test methods for determining whether each test assertion is met. The test methods of this Annex correspond 1:1 to the assertions of Annex A, and the conformance determination (Clause 11) is based on the records of tests performed in accordance with the procedures of this Annex.

> NOTE — The **pitfall** entry of each test method describes the paths by which that test itself can be wrongly passed. Before performing a test, the tester reads that entry and uses it to confirm that the tester's own test does not fall under those paths.

C.0 Common rules

C.0.1 Scope

For each assertion specified in the body of this specification, this Annex specifies the test method for determining whether that assertion is met. This Annex specifies the types of test, the minimum conditions for a test to be established, the mandatory entries of the test record, the conditions under which a test result becomes invalid, and the re-evaluation scope upon revision of a test method. This Annex does not specify criteria for assigning grades to test results, qualification requirements for certification bodies, detailed specifications of test facilities or the manner in which test tools are implemented. Nor does this Annex specify which design the test target system shall adopt; it specifies only how it is confirmed that the adopted design meets the assertions. The 72 individual test methods specified in C.1 to C.7 shall follow the common rules of this clause (C.0), and where an individual test method conflicts with the provisions of C.0, the provisions of C.0 take precedence.

C.0.2 Three types of test method

Test methods are classified into three types: injection, demonstration and audit. An **injection test** is a test in which the tester deliberately produces in the test target a condition that an assertion prohibits or requires to be detected, and observes whether the test target detects, blocks or records it. A **demonstration test** is a test in which the test target is operated under normal conditions to confirm that the required function is performed in an observable form; it is applied only to assertions for which a defect condition cannot be produced or for which producing it is not safe. An **audit test** is a test in which the records, configurations and documents produced by the test target are reviewed to confirm that the assertion is met; it is applied where the subject of the assertion is not an individual run but a property of outputs over a period. An assertion designated for automated check shall in principle be tested by an injection test. The reason is that ‘the existence of a checker’ is not ‘the operation of a checker’ — auditing only whether a checker is installed, its configured values and whether its call path is registered cannot confirm whether that checker fires on an actual defect, and failure modes in which a checker is silently deactivated or bypassed through an exception path are not observed by audit. Where an assertion designated for automated check is tested by a type other than injection, the reason why injection is not possible shall be entered in the test record. The ‘checking method’ of Annex A can indicate in combination the means employed in the conformance determination (for example, ‘automated+audit’), but the ‘test type’ of this Annex designates only one for each assertion. For an assertion with a combined indication, the remaining means are included as steps within the procedure of that assertion.

C.0.3 Test record

Every test shall record the following entries: the test date, the version and identifier of the test target, the version of the test method applied, the tester, the test result (conforming,

nonconforming or inconclusive), and the observational evidence supporting that result. The observational evidence shall be not the tester's conclusions but facts observed during the test — outputs, log entries, measurement values, screen recordings and the like — and shall be preserved in a form from which a third party can reach the same conclusion from the same evidence. A result shall be recorded even where it is ‘inconclusive’, and an inconclusive result shall not be treated as conforming. A test without a record is deemed not to have been performed. Test records shall be preserved for as long as the conformance determination concerned remains valid.

> NOTE — Correspondence of result vocabularies. This specification uses three vocabularies at different levels — for the SUT verdict (Clause 7), pass, fail and not-measured; for the result of an assertion test (this clause), conforming, nonconforming and inconclusive; and for the Implementation Conformance Statement (ICS, 11.3), met, not met and not applicable. The correspondence between the results of assertion tests and ICS declarations is as given in Table C.0.3-1 (informative). ‘Not-measured’ as used by an individual test method for a sample or a test step is not the not-measured of the SUT verdict (3.9) but means that the basis for a verdict was not established for that sample or step, and its treatment follows the pass criteria of each test method.

Table C.0.3-1 (informative) — Correspondence between assertion test results and ICS declarations

Test result (C.0.3)	ICS declaration (11.3)	Basis
Conforming	Met	A declaration of met is supported by the test record of that determination (C.7.2)
Nonconforming	Not met	—
Inconclusive	Not met — not declared as met	An inconclusive result is not treated as conforming (C.0.3) · not applicable is limited to a mismatch of the applicable class (**PAVS-CNF-02**)
Pending re-evaluation (C.0.6)	Not declared as met	The pending-re-evaluation state is not indicated as conforming (C.0.6)
Not a test target (the applicable class differs from the claimed class)	Not applicable — reason stated alongside	**PAVS-CNF-02**

C.0.4 Discipline of sample audits

Where, in an audit-type test, the population is not reviewed in its entirety, the tester shall record the sample size, the sampling method, the period to which the sample belongs and the size of the population. The sample shall be selected in accordance with a rule set in advance — random sampling, systematic sampling, or stratified sampling of high-risk strata designated by the specification — and shall not be chosen arbitrarily by the tester. Where the test target organization presents a sample, that sample shall be replaced or supplemented by a sample drawn separately by the tester. Where the sample is skewed toward a particular period or a particular component of the population, the fact of that skew and its reason shall be recorded, and a conforming determination for the population as a whole shall not be made from a skewed sample. Where even 1 nonconformity is observed in a sample audit, the assertion concerned shall be determined to be nonconforming.

C.0.5 Invalidity conditions of tests

The result of a test that falls under any of the following cases is deemed invalid, and an invalid test shall not be used as the basis for a conforming determination. First, where the population, input range or test interval has been reduced in order to make the test pass. Second, where, in an injection test, the test target system or its operator knew in advance the content, timing or location of the defect to be injected, so that a response limited to that defect was possible. Third, where, in a test using a control, the control itself has not been independently verified — an unverified control can establish a defect of the test target as normal. Fourth, where the test tool shares the same components or the same assumptions with the test target, so that an error in those assumptions appears identically in the test tool and in the test target. Fifth, where the criteria for determining the test result were changed after the test was performed. Where the organization that developed the test target performs the test itself (self-test), the tester shall prove by records that none of the above conditions applies, and a self-test without such proof is deemed invalid.

C.0.6 Revision of test methods

Where a test method is revised, the revised version shall also specify the re-evaluation scope of existing conformance determinations made under the test method before revision. The re-evaluation scope shall be designated as one of three grades — ‘no retest required’, ‘retest of the assertion concerned only’ and ‘retest of the entire related assertion group’ — and a revision without such designation is deemed ‘retest of the assertion concerned only’. Where a revision of a test method changes the strictness of the determination criteria — where thresholds, sampling discipline or invalidity conditions are changed — it shall not be designated ‘no retest required’. The version of a test method shall be incremented at every revision, and the test record shall state the version of the test method applied (C.0.3). Until the retest is completed, the existing determination shall be indicated as ‘pending re-evaluation’, and the pending-re-evaluation state shall not be indicated as conforming.

C.1 ITM — Item test methods (10)

C.1.1 PAVS-ITM-01 — Confirms that, if test logic or data exist outside the scenario description, the scoring system rejects them at the loading stage, and that, in an environment without auxiliary code, a third party reproduces the same verdicts and the same measurement coverage from the same description alone.

Test type: injection

Preconditions

- 2 or more scenarios with a verdict history are registered in the corpus, and the list of all scenarios in the corpus can be queried.
- The path that checks the external dependencies of a scenario runs within the production execution path, and its result determines whether execution starts.
- A third party not involved in the authoring or operation of the scenarios concerned is designated as the reproducing executor, and the third-party environment can be built from only the dependencies declared by the distribution package.

Procedure

1. Taking all scenarios in the corpus as the population, the tester draws 2 at random (recording the population count and the seed). The test-target scenarios are not nominated by the range operator.

2. For each drawn scenario, the tester makes 4 copies and injects a different external dependency into each: ① a direct call to auxiliary code outside the scenario; ② an identifier

pointing to a constant built into the scoring system code; ③ a reference to external data not declared in the scenario description; ④ a late-bound call resolved through a string identifier at execution time.

3. The tester feeds the injected copies into the same loading and execution path as production, and observes acceptance or rejection at the loading stage, the verbatim rejection reason and whether execution outputs, item verdicts and results documents are generated for the rejected copies.

4. The tester feeds 2 kinds of placebo control into the same path: (a) the original without injection; (b) a harmless variant copy in which the dependencies are left unchanged and only comments, formatting and internal identifier names are changed.

5. The tester passes only the original scenario description and the execution pinning information (version, seed and parameters) to the third party, has the third party execute it in an environment without auxiliary code or internal material, and then receives the configuration list of that environment as well.

6. The tester compares the per-item verdicts and the measurement coverage list of the third-party run with those of the original run entry by entry, and executes the late-binding-injected copy in the same third-party environment and observes the result.

Pass criteria

- All 4 kinds of injection are rejected at the loading stage, the kind of dependency injected is identified in each rejection record, and no verdicts or results documents are generated for the rejected copies. Where the late-binding injection is not caught at loading, the run in the third-party environment shall abort, stating the absence of the dependency, and if it produces a verdict, the result is a fail.
- Neither of the 2 kinds of placebo control is rejected, and both execute normally. If the harmless variant copy is rejected, the check is looking not at external dependencies but at the fact of change itself, and the result is therefore a fail.
- The per-item verdicts of the third-party run agree with the original run for all items, the sets of measurement coverage entries are identical, and the third-party environment configuration list contains no auxiliary code or internal material.

Record items

- Population count, draw seed and target scenarios, the content of the 4 kinds of injection, and the acceptance or rejection result and verbatim rejection reason for each
- Execution results of the 2 kinds of placebo control (original and harmless variant)
- Identification of the third party and the basis for its non-involvement, and the third-party environment configuration list, pinned seed and version
- Comparison table of per-item verdicts and measurement coverage between the original run and the third-party run

> Caution — **pitfall**: The case in which the loading check rejects every ‘copy that differs from the registered version’ wholesale, so that all injections are rejected while external dependencies are in fact not examined at all, and the case in which a late-bound reference passes the format check and is recorded as ‘0 external dependencies’ — unless a harmless-variant placebo and a run in a third-party environment without auxiliary code are included as well, both paths pass this test.

C.1.2 PAVS-ITM-02 — Confirms that every item is linked 1:1 to a check that is actually executed and fires by looking at values, and that an item that has only a mapping but is not executed, or that always returns true, is exposed not as a pass but as not-measured or as a defect.

Test type: injection

Preconditions

- The item list and the item-to-check mapping exist in machine-readable form, and the items of the distributed corpus and of all profiles are included in that list.
- A per-item check execution record (whether executed, verdict produced and basis value) is left in the verdict output.
- A copy in which, item by item, the check wiring is cut or the measurement input values are replaced can be executed through the same path as production.

Procedure

1. Taking as the denominator all items registered in the distributed corpus and in all profiles, the tester performs a mapping audit and counts the unmapped, many-to-one and one-to-many mappings and the items not executed in any profile. The tester does not reduce the denominator to recent runs or to the current profile.

2. From the items whose mapping has been confirmed, the tester draws 5 by stratified random sampling. The tester draws at random 2 from the group of items for which only passes have been observed in the history of the most recent issue cycle, and 3 from the remaining group, and records the population counts and the seed (if a group has fewer items than its allocation, all items of that group are included and the shortfall is made up from the other group; all items, if there are fewer than 5).

3. For each sampled item, the tester executes the test with a copy in which the wiring is cut so that only that item's check is not executed, and observes the verdict of that item and how it is counted in the denominator and numerator of the pass rate, and whether the verdicts of the remaining items not cut are the same as in the run before cutting.

4. For each sampled item, the tester executes a copy in which the measurement value referred to by the check is replaced with a value on the opposite side of the threshold (value-inversion injection), and observes whether the verdict flips.

5. The tester injects into the item list 1 item that has a name but is linked to no check, attempts its registration and execution, observes the result, and then withdraws the injected entry.

Pass criteria

- There are 0 unmapped items, the many-to-one and one-to-many mappings agree with the declarations in the item definitions, and the items not executed in any profile are produced as a list.
- In all 5 wiring cuts, the verdict of the item concerned is produced as not-measured and is counted as neither a pass nor a fail, and the verdicts of the remaining items not cut are identical to those of the run before cutting.
- In all 5 value-inversion injections, the verdict flips pass↔fail, and the registration or execution of the item not mapped to a check is rejected with the absence of a check mapping as the reason.

Record items

- Definition of the denominator (scope of corpus and profiles) and its count, tally table of the mapping audit, and audit time and item list version
- Stratified draw seed and sample list, the verdicts observed after cutting, and the result of the identity comparison of the verdicts of the remaining items
- Replacement values of the value-inversion injection and the observed verdict changes
- Result of the injection of the item without a check mapping, and the rejection reason

> Caution — **pitfall**: In an implementation in which cutting the wiring aborts the entire run so that all items become not-measured at once, the cutting test passes on seeing only ‘not-measured produced’, but whether that item's check looks at values remains unconfirmed — unless the identity of the remaining items' verdicts and the value-inversion injection are required as well, a

> check that always returns true passes both ‘100% mapping’ and ‘not-measured when cut’.

C.1.3 PAVS-ITM-03 — Confirms that, if a value differing from the physical quantity, unit or measurement provenance required by an item enters scoring, it is treated not as a pass or fail but as not-measured and the reason is recorded, and that, conversely, for values that agree with the declaration, normal verdicts are produced in both directions.

Test type: injection

Preconditions

- The required physical quantity and unit and the required measurement provenance of each item are declared in the item definition.
- The verdict output has a per-item not-measured reason field, and the reason is identified by a code.
- Values can be injected, in the same format as in production, at the boundary at which the measurement channel passes values to the scoring engine.

Procedure

1. The tester selects 3 items that include a threshold comparison and require mutually different physical quantities (EXAMPLE: an item requiring force, an item requiring velocity and an item requiring distance).

2. For each item, the tester injects four kinds of ineligible value in turn at the measurement channel boundary: ① a value of a different kind of physical quantity but of a magnitude that yields a pass if compared with the threshold as it is; ② a value of the same physical quantity but in a different multiple unit (a milli-unit figure in the place of the base unit); ③ a value whose field name is the same as the required physical quantity but whose actual physical quantity is different; ④ a value whose physical quantity, unit and name all agree but whose measurement provenance is a commanded value or a default value.

3. For each injection, the tester observes the item verdict, the not-measured reason code, and the denominator and numerator of the pass rate.

4. The tester executes a placebo control for each of the 3 items. The tester injects values whose physical quantity, unit and provenance agree with the declaration, setting their magnitudes so that at least 1 exceeds the threshold and at least 1 falls short of the threshold.

5. The tester queries the measurement provenance record of the basis values from which the verdicts were produced, and confirms whether it is entered as the same as the injected provenance.

Pass criteria

- In all 12 ineligible injections (3 items × 4 kinds), the verdict is produced as not-measured, there are 0 cases counted as a pass, and each not-measured reason code points to the mismatch of physical quantity, unit or provenance (it is not replaced by another reason such as out of range or outlier).
- All 3 placebo controls produce normal verdicts rather than not-measured, and of these at least 1 is a pass and at least 1 is a fail.
- Not-measured cases are included in neither the denominator nor the numerator of the pass rate (**PAVS-VDT-01**, C.3.1) and, without being deleted from the aggregate, are counted separately as not-measured, and each not-measured record states both the required physical quantity, unit and provenance and the physical quantity, unit and provenance of the value that actually entered.

Record items

- Required physical quantity, unit and provenance of the injection-target items, and the physical quantity, unit, magnitude and provenance of the injected values
- Verdict for each injection and the verbatim not-measured reason code
- Verdicts of the 3 placebo controls and their values relative to the threshold
- Changes in the denominator and numerator of the pass rate

> Caution — **pitfall**: An implementation that produces ‘always not-measured’ for ineligible values passes as it is if there is only 1 placebo control or if only one direction is tested, and such an implementation, in actual testing, folds even normal measurements into not-measured and silently empties the pass-rate denominator — unless placebos in both the pass and the fail direction are required for every item, ‘the unit check is strict’ cannot be distinguished from ‘no scoring takes place at all’.

C.1.4 PAVS-ITM-04 — Confirms that a metric used for verdicts or monitoring is actually produced in every kind of test to which it applies, that in a kind of test in which it is not produced it is exposed as missing rather than as 0 or a default value, and that a constant series is not counted as measurement assurance.

Test type: injection

Preconditions

- A list exists in which the applicable kinds of test of each metric are declared.
- Two or more kinds of test are in operation, and all execution outputs of the most recent 1 issue cycle can be queried.
- A copy in which the output of a measurement channel is blocked in a particular kind of test can be executed through the same path as production.

Procedure

1. The tester builds a matrix of metrics × applicable kinds of test and, covering all runs of the kind of test concerned in the most recent 1 issue cycle (runs are not selected), classifies the recording state of each cell as value present, missing, all 0 or constant series (variance 0 across all runs).

2. For each cell classified as constant series, the tester executes 1 test that creates a condition under which that metric is expected to move, observes whether the value moves and, if it does not, reclassifies that cell as assurance unconfirmed.

3. From the cells classified as value present, the tester draws 3 cells at random (all, if there are fewer than 3 cells) (recording the seed), blocks the output of the measurement channel of that metric in the kind of test concerned and executes the test (missing-data injection).

4. In the missing-data-injection run, the tester observes whether the metric concerned is recorded as missing or replaced with 0 or a default value, what the verdict of the items using that metric becomes, and whether a monitoring rule aborts the run.

5. The tester executes, in the same kind of test, a condition under which the same metric actually becomes 0 (measured-0 control), and confirms whether that record is distinguished from the missing record in the output.

6. Among the declared applicable kinds of test, the tester lists the cells with no output history at all and the cells reclassified as assurance unconfirmed because they failed the operation check.

Pass criteria

- In all 3 missing-data injections, the metric is recorded as missing, there are 0 cases replaced with 0 or a default value, and the verdict of the items using that metric is produced as not-measured.

- The measured-0 control is recorded not as missing but as the value 0, so that the two states are distinguished in the output.
- Of the constant-series cells, only those that passed the operation check are counted as measured, and the cells that did not pass and the cells with no output history are produced and made public as the assurance-unconfirmed list (if there are 0, 0 is entered).

Record items

- Metric × kind-of-test matrix, recording state of each cell and the number of runs used for the tally
- Content of the operation-check runs for constant-series cells and the observed value movements
- Target metrics and kinds of test of the missing-data injection and the draw seed, and the observed recording form, item verdicts and whether monitoring aborted the run
- Recording form of the measured-0 control, and the list of assurance-unconfirmed cells

> Caution — **pitfall**: The case in which a metric always produces 0 in a kind of test to which it does not apply and the matrix tally counts this as ‘value present’ — a constant 0, being neither missing nor measured, combines with a monitoring threshold to abort normal tests erroneously, and unless a condition under which the metric actually moves in that cell is executed even once, measurement assurance in the tally remains at 100%.

C.1.5 PAVS-ITM-05 — Confirms that the discriminating-power classification of items is computed and recomputed with only the SUT test history as the denominator, and that items for which only a single verdict has been observed and items with insufficient samples are indicated as boundary unverified and unclassifiable, respectively.

Test type: injection

Preconditions

- The verdict history of each item is preserved per test, and each run carries a label of SUT test, negative control or self-check so that it can be queried.
- The discriminating-power classification rule (definition of the denominator, minimum number of runs and minimum number of observations per verdict), its output (the per-item classification table) and the recomputation triggers are declared.
- Without tampering with the operational history, the classification can be recomputed on a test copy of the history.

Procedure

1. The auditor obtains the current classification table, recomputes the classification without using the range operator's computation tool, on the basis of only the raw verdict-history data and the classification rule definition, and compares the results item by item.

2. The tester obtains the list of runs actually included in the denominator of the classification and counts them by label (SUT test, negative control, self-check and not-measured).

3. The tester injects synthetic runs into the test copy of the history: ① adding 1 failing run to 1 item classified as constant-pass; ② removing all failing runs from 1 item classified as discriminating; ③ adding 1 run in which the qualification requirement of the item concerned does not hold.

4. The tester activates the declared recomputation trigger and observes whether the classifications of the three items move to discriminating, constant-pass and not qualified, respectively, and whether the triggering occurred automatically or required manual operation.

5. For all items classified as constant-pass or constant-fail and all items below the minimum count, the tester confirms whether the boundary-unverified or unclassifiable indication appears in both the item definition and the results document.

6. The tester compares the time of the last recomputation with the time of the latest verdict history and confirms whether the interval exceeds the declared issue cycle.

Pass criteria

- The result of the independent recomputation agrees with the current classification table for all items, the runs included in the denominator consist only of runs labeled SUT test (0 negative controls or self-checks), and not-measured runs are counted as neither a pass nor a fail.
- In each of the 3 injections, the classification moves in the expected direction and the classifications before and after the move are left in the output record, and items falling short of the minimum number of runs or the minimum number of observations per verdict are classified as none of discriminating, constant-pass or constant-fail.
- All constant-pass, constant-fail and unclassifiable items carry the corresponding indication, and the time of the last recomputation is within the declared issue cycle.

Record items

- Item-level comparison table of the current classification table and the result of the independent recomputation, and the definition of the denominator and the number of runs by label
- Content of the 3 injections, the observed classification moves and the triggering mode (automatic or manual)
- List of items missing the indication, and the classification state of the items below the minimum count (if there are 0, 0 is entered)
- Time of the last recomputation, declared issue cycle and time of the latest history

> Caution — **pitfall**: The case in which negative controls or self-check runs of the scorer are mixed into the classification denominator, so that a tautological item that has never once produced a fail in SUT tests is classified as ‘discriminating’ and its boundary-unverified indication is stripped away — a fail of a control is evidence that the scorer fires, not evidence that the item discriminates between SUTs, and unless the denominator is counted by label, the two pieces of evidence are merged into one number.

C.1.6 PAVS-ITM-06 — Confirms that the boundary verification of items for which only a single verdict has been observed was performed with reverse-direction controls in which the stimulus actually occurred, and that there are no cases in which discrimination was created by adjusting the threshold.

Test type: audit

Preconditions

- The boundary verification history and the change history of item thresholds are each preserved together with time, actor and reason.
- The list of constant-pass and constant-fail items and the per-item boundary-verification-complete indication (output of **PAVS-ITM-05**) exist.
- The outputs of control runs (per-item verdicts, basis values, measurement provenance and stimulus metrics) can be queried.

Procedure

1. The tester compares all items indicated as boundary verification complete with the boundary verification history, and counts the ‘complete’ indications without a corresponding record.

2. Taking as the population the boundary verification history of the most recent 2 issue cycles (the most recent 12 months, if no cycle is declared), the tester draws 10 at random (all, if the population is fewer than 10). The tester records the population count and the draw seed.

3. For each sample, the tester diffs the original scenario against the control scenario entry by entry, and confirms whether the changed entries are limited to one, the direction of the stimulus (sign, order, target position and the like). For a control whose derivation relationship is not confirmed by the history, the tester compares all entries. A case in which the stimulus has been removed, or in which the threshold, verdict criteria or measurement target has been changed as well, is counted as nonconforming.

4. For each sample, the tester queries the outputs of the control run and confirms that the verdict of the designated item is a fail, that the measurement provenance of the basis value is measured and the value lies on the opposite side of the threshold, and that a measured value of the stimulus metric exists. A case in which the verdict is not-measured or the stimulus metric is missing is counted as nonconforming.

5. The tester obtains the entire threshold change history for the same period, lists the cases in which ‘the threshold of an item without discrimination was changed before or after boundary verification’, and confirms whether the reason for each case cites one of a clause of an external reference standard, the manufacturer's rated specification or the identifier of a measured record.

6. The tester injects 1 item definition whose threshold has been changed in the direction that creates discrimination into the registration path of a test account or a copy path, observes whether the change history record and the demand for a reason and approval fire, and then withdraws the injected entry.

Pass criteria

- There are 0 ‘boundary verification complete’ indications without a corresponding record, and in all 10 samples the entries changed relative to the original scenario are limited to one, the stimulus direction (0 cases of stimulus-removal controls or of accompanying changes of threshold or criteria).
- In the control runs of all 10 samples, the designated item produces a fail with a basis value of measured provenance, and a measured value of the stimulus metric exists.
- There are 0 cases in which the threshold of an item without discrimination was changed in temporal association with boundary verification, or each such case cites a physical basis of at least one of the above three kinds, and the injection of a threshold change without a reason is not registered as it is.

Record items

- Population definition, count, draw seed and sample list, and the result of the comparison of ‘complete’ indications
- For each sample, the result of the diff between the original scenario and the control, and the nonconforming count
- For each sample, the verdict, basis value and measurement provenance of the designated item and the measured value of the stimulus metric in the control run
- Comparison table of the threshold change history and the boundary verification times, and the observed result of the threshold-change injection

> Caution — **pitfall**: The case in which a control is in the reverse direction on paper but the stimulus does not occur in the actual run, so that it is registered as ‘boundary verification complete’ while the designated item is not-measured or the stimulus metric is missing — if only

> the description documents are compared and the basis values and measurement provenance of the control run outputs are not opened, a fail arising from absence of stimulus hardens into evidence of discriminating power, and the threshold of that item is never verified.

C.1.7 PAVS-ITM-07 — Confirms that the distribution package of the standard item set is not a list of documents but a corpus that is equipped with scenarios, controls and expected verdicts and that, executed in a new environment in exactly the form the recipient receives it, reproduces the issued verdicts.

Test type: audit

Preconditions

- The version and distribution path of the distributed corpus are declared.
- The list of items included in the corpus and the item set version cited by the current results documents can be queried.
- The auditor can newly build an execution environment from only the environment requirements declared by the distribution package.

Procedure

1. The tester obtains the distribution package from the distribution path in exactly the form the recipient receives it. A copy in the internal operational repository is not taken as the audit target.

2. The tester compares the item list of the distribution package with the currently issued item set and lists the undistributed items, and compares the version of the distribution package with the item set version cited by the results documents.

3. Taking all items of the corpus as the population, the tester draws 10 at random (all, if there are fewer than 10 items), and for each item confirms the actual existence of the four components — scenario description, control scenario, expected-verdict declaration and execution pinning information (version, seed and parameters). The tester records the population count and the draw seed.

4. The tester confirms whether the expected-verdict declaration of each sample contains the per-item verdict values and their basis thresholds, physical quantities and units, and whether the control differs from the original scenario in only one entry, the stimulus direction, and counts the cases that are empty or ‘not applicable’.

5. In an environment newly built from only the environment requirements declared by the distribution package, the tester executes 3 samples (at least 1 of them a control) without any material other than the distribution package, and records the environment configuration list.

6. The tester compares the verdicts produced by the 3 runs with the expected verdicts, and confirms whether the designated item is produced as a fail in the control run.

Pass criteria

- In all 10 samples, the four components actually exist and there are 0 cases in which the expected verdict is empty or ‘not applicable’, and there are 0 undistributed items or their list is indicated in the distribution package.
- The version of the distribution package agrees with the item set version cited by the results documents.
- The 3 runs are executed in the newly built environment from the distribution package alone, the produced verdicts agree with the expected verdicts, and the designated item of the control run is produced as a fail.

Record items

- Corpus version, acquisition path, acquisition time and acquisition form, and the result of the comparison with the version cited by the results documents
- Population count, draw seed and sample list, and the table of existence of the four components for each sample
- List of undistributed items and the number of missing expected verdicts (if there are 0, 0 is entered)
- Configuration list of the new environment, and the result of the comparison of the verdicts produced by the 3 runs with the expected verdicts

> Caution — **pitfall**: The case in which the distribution package is executed only in the range operator's existing environment and recorded as ‘executed from the distribution package alone’, and the case in which the distributed version lags behind the version cited by the results documents so that the issued verdicts cannot be reproduced from the corpus the recipient received — if execution takes place in an internal environment without comparing the acquisition form and version, the corpus passes this audit even if it is in fact no more than a list of documents.

C.1.8 PAVS-ITM-08 — Confirms that a performance or safety baseline cannot be registered or cited without the values and units of its condition axes, that the declared condition values agree with the effective values of the actual runs, and that comparison between baselines with different conditions is blocked.

Test type: injection

Preconditions

- Condition axis fields (entry, value and unit) are declared in the baseline definition, and all current baselines can be queried in machine-readable form.
- The paths for registering and citing baselines can be executed in the same way as in production.
- The effective values of the condition axes can be queried from the test execution records that produced each baseline.

Procedure

1. The tester injects into the registration path 1 baseline definition with the condition axis fields left empty, 1 definition that has the fields but whose value is ‘not applicable’, and 1 definition whose value is a free-text description without a unit (EXAMPLE: ‘standard conditions’), and observes acceptance or rejection and the rejection reason. If any injection is accepted, the tester confirms whether that baseline can be cited in an actual verdict and then withdraws the injected entries.

2. The tester machine-audits all current baselines and counts the cases in which the condition axis fields are missing, ‘not applicable’ or a free-text description without a unit (all cases, not a sample).

3. The tester has two baselines with mutually different condition values cross-cited in the verdict of the same item, and observes whether a comparison block or a condition-mismatch indication fires.

4. The tester draws 10 at random from all current baselines (all, if there are fewer than 10 in total) (recording the population count and the seed), and checks whether the condition axes declared by each baseline include the sensitive axes of the item definition concerned, and whether the declared condition values agree with the effective values of the run that produced that baseline.

5. For 1 of the samples, the tester performs, with the same SUT, 2 runs in which the condition value is changed to each end of the declared range of application, and 3 repeated runs with the condition fixed.

6. The tester compares the difference in measured values between the runs at both ends of the condition with the max–min range of the 3 repetitions, and observes whether the verdict shifts.

Pass criteria

- All 3 injections are refused registration, the missing or ineligible fields are stated in the rejection reason, and in the audit of all current baselines there are 0 cases of missing, ‘not applicable’ or unit-less free-text condition axes.
- In the cross-citation of baselines with different conditions, verdict production is blocked or the condition mismatch is indicated in the result.
- In all 10 samples, the declared condition axes include the sensitive axes of the item definition and the declared values agree with the effective values of the runs, and in the 1 measured confirmation the difference in measured values between both ends of the condition exceeds the max–min range of the 3 repetitions under identical conditions.

Record items

- Content, registration result and verbatim rejection reason of the 3 injections, and the tally of the audit of all current baselines
- Observed result of the cross-citation attempt
- Population count, draw seed and sample list, the result of the sensitive-axis comparison and the result of the declared-value versus effective-value comparison
- Measured values and verdicts of the runs at both ends of the condition, and the dispersion of the 3 repetitions under identical conditions

> Caution — **pitfall**: The case in which the condition axes have values and units and pass the format check, but those values are not actually applied in execution (as when the declared load and the effective load differ), so that the baseline is produced under conditions it did not declare — if only the field format is audited, a state remains in which the same SUT both passes and fails on the same task, and unless the shift width of runs with changed condition values is compared with the repetition dispersion, measurement noise is recorded as ‘sensitive axis confirmed’.

C.1.9 PAVS-ITM-09 — Confirms that the item set includes systematic variation along perturbation axes such as object placement, camera viewpoint, initial state and instruction variants, and that the variation is implemented not as random repetition but as differences in execution conditions at least as large as the declared variation width.

Test type: audit

Preconditions

- The item set declares the list of perturbation axes, the variation levels of each axis and the variation width between levels.
- The values of the applied variation conditions are left in each test execution record.
- Records of repeated runs under identical conditions, or measurement repeatability data, can be queried.

Procedure

1. Taking the entire item set as the denominator (including items excluded from profiles, but indicated separately), the tester builds an axis × item matrix and counts the cells in which

variation is defined, the single-condition cells and the cells in which condition values are not recorded.

2. Taking all items for which variation is declared as the population, the tester draws 8 at random (all, if the population is fewer than 8). The tester records the population count and the draw seed.

3. For each sample, the tester queries the test records executed at mutually different variation levels and compares, axis by axis, whether the condition values left in the records actually differ from level to level. A case in which the values are the same or are not recorded is counted as ‘declared only, not applied’.

4. The tester confirms whether the difference in condition values between levels is at least the declared variation width, and whether that difference arose on an axis other than the random seed (object placement, camera viewpoint, initial state or instruction). Repetitions that differ only in seed are not counted as variation.

5. For each sample, the tester tallies the verdict distribution by variation level (counts of pass, fail and not-measured), and lists the items whose verdict is the same at all levels as ‘perturbation discrimination unconfirmed’.

6. The tester queries the items newly established in the most recent 1 issue cycle, and confirms the items registered without a perturbation axis declaration and whether that fact is indicated in the item set document.

Pass criteria

- In all 8 samples, execution records at mutually different variation levels exist and the recorded condition values differ from level to level by at least the declared variation width, and the levels consisting of repetitions that differ only in seed and the ‘declared only, not applied’ cases number 0.
- The axis × item matrix with all items as the denominator, the verdict distribution by level and the perturbation-discrimination-unconfirmed list are produced and made public in the item set document.
- For newly established items without a perturbation axis declaration, that fact is indicated in the item set document.

Record items

- List of perturbation axes, levels of each axis and declared variation width, and the axis × item matrix (including single-condition cells and cells with unrecorded condition values)
- Population count, draw seed and sample list
- For each sample, the execution condition values, the result of comparing the differences between levels with the variation width, and the verdict distribution by level
- List of items declared only but not applied, the perturbation-discrimination-unconfirmed list and the list of newly established items without a perturbation axis declaration

> Caution — **pitfall**: The case in which the variation is implemented as repetitions that change only the random seed, or as minute changes whose condition-value differences fall within measurement repeatability, and is tallied as ‘N-level variation’ — unless condition values are compared axis by axis and the differences are checked against the declared variation width, repetitions of the same condition masquerade as evidence of generalization capability, and the verdict distribution is then produced as ‘pass at all levels’ and mistaken for strong evidence.

C.1.10 PAVS-ITM-10 — Confirms that, for every published rated capability, an item exists that confirms the sustainability of that capability, that a single success is not used as the basis for being met, and that limits not reproduced by the simulation are indicated as physical-hardware-only.

Test type: audit

Preconditions

- A list of the published ratings of the SUT (payload, speed, continuous operation and the like) exists as a document.
- For each item of the item set, the correspondence indicating which rating that item targets is recorded.
- The termination conditions (time, number of repetitions and abort reasons) of sustained-test items are declared.

Procedure

1. The tester checks whether there is an item corresponding to each entry of the published rating list. The tester enumerates the ratings without a corresponding item.

2. The tester checks the termination conditions of the corresponding items. An item that terminates on a single success alone is not counted as a sustained-test item.

3. For those sustained-test items that are performed in simulation, the tester confirms whether their limiting factors (heat, current, wear and the like) exist in the model used. If they do not exist, the tester confirms whether that item is indicated as physical-hardware-only.

Pass criteria

- A corresponding item exists for every published rating.
- None of the corresponding items terminates on a single success.
- Items that depend on limiting factors absent from the model are indicated as physical-hardware-only.

Record items

- Correspondence table of the rating list and the items (including ratings without a corresponding item)
- Termination conditions and actual termination reason of each sustained-test item
- Basis for the physical-hardware-only indication (confirmation that the factor is absent from the model)

> Caution — **pitfall**: If a rating is read as ‘the maximum value achievable once’, this test always passes — the meaning of a rating is the **sustained working capability** under the declared conditions. Also, a sustained-test item passing in simulation can be due to the limiting factor being **absent** from the model, so before a pass is taken as a basis, the existence of the factor is confirmed first.

C.2 MEA — Measurement test methods (8)

C.2.1 PAVS-MEA-01 — Confirms that every value used in scoring carries its measurement provenance in machine-readable form, that an item requiring measured values is not scored with values of non-measured provenance, and that the ‘measured’ declaration itself has not been filled with command-family signals.

Test type: injection

Preconditions

- The measurement provenance vocabulary is declared as a machine-readable enumeration that distinguishes measured, commanded value, default value and estimate from one another, and the determination rules for each category exist as a document.
- The complete list of items requiring measured values and the measurement keys referred to by each item can be queried.
- The scoring engine can be re-executed on arbitrary input through the same entry path as in production, and an evaluation trace enumerating every value referred to by that run can be obtained.

Procedure

1. The tester performs 1 normal test on the production path, fixes the population not as ‘the values displayed in the results document’ but as ‘every value referred to by the evaluation trace of the scoring engine’ and records its count, counts for each value whether a provenance field is present, and, if any value has been excluded from the population, records the reason for the exclusion alongside.

2. The tester takes every item requiring measured values as the target — if there are more than 20, the target is 20 items, consisting of the 10 most cited in recent results documents and 10 drawn at random from the rest — and, for each item, makes copies in which only the provenance of the keys referred to by that item is changed to ‘commanded value’, ‘default value’ and ‘estimate’, respectively, and a copy from which the provenance field is removed — in all of them leaving the values themselves unchanged — and re-executes them.

3. As placebo controls, the tester re-executes through the same path (a) a copy in which the provenance is rewritten from ‘measured’ to ‘measured’ and (b) a copy in which only the provenance of keys not referred to by the item concerned is changed to non-measured, and confirms whether the verdicts of the two copies are identical to those of the original run.

4. The tester compares the verdict and reason code of each copy with those of the original run, and distinguishes whether the ineligibility handling names ‘ineligible provenance’ as the reason or whether the copy was rejected for another reason, such as a copy format error or an input validation failure.

5. Taking all of the keys declared as measured that are referred to by items requiring measured values, the tester traces their production paths backward and counts the keys that are filled with setpoint values, integrals of setpoints, constant multiples of setpoints or configured constants.

6. Where a measurement key of the same name is used in two or more kinds of test, the tester queries the provenance declaration for each kind and counts the disagreements.

Pass criteria

- Among the values referred to by the evaluation trace, there are 0 values without a provenance field, and 0 values have been excluded from the population.
- In every non-measured-label copy and every provenance-removed copy, the item concerned is not produced as a pass but is handled as not-measured or as test rejection, with ineligible provenance stated in the reason, and the two placebo-control copies produce verdicts identical to those of the original run.
- There are 0 keys that are referred to by items requiring measured values and whose production path is command-family or a configured constant, and there are 0 provenance disagreements between kinds of test for the same key name.

Record items

- Enumeration values of the provenance vocabulary, the determination rules for each category, and the count and excluded entries of the evaluation-trace population
- Comparison table for each copy (injected provenance, verdict produced, reason code) and the verdicts of the 2 placebo-control copies
- Count of values without a provenance field and the list of their keys

- Results of the backward trace of the production paths of keys declared as measured (count of keys determined to be command-family) and the results of the provenance comparison across kinds of test

> Caution — **pitfall**: If the re-execution path rejects ‘modified copies’ wholesale at the format-check stage, none of the injected copies produces a pass and the test passes, yet the provenance gate never fires even once; unless it is confirmed whether the placebo controls give the same verdicts as the original run, the absence of the gate is recorded as the operation of the gate.

C.2.2 PAVS-MEA-02 — Confirms that missing data is not replaced with 0 or a default value at the measurement record layer, and that an input whose decisive key is missing produces a pass on no verdict surface.

Test type: injection

Preconditions

- Every verdict surface (item evaluation entry point) of the scoring system can be enumerated, and the list of items evaluated in the recent period can also be queried.
- The measurement record schema has a representation of missing (‘none’) that is distinct from the value 0.
- A verdict surface can be invoked on arbitrary input through the same entry path as in production.

Procedure

1. The tester enumerates every verdict surface, fixes them as the population at the time of the test and records the count, checks to which verdict surface of this population each item evaluated in the recent period corresponds and counts the items that have no corresponding verdict surface, and, if any verdict surface has been excluded, records the reason for the exclusion.

2. The tester feeds an input in which every field is missing into each verdict surface of the population through the production entry point as is, and collects the returned verdicts.

3. Into the same verdict surfaces, the tester feeds a partially missing input in which only the decisive key of that item is missing and the remaining fields have normal values, and collects the returned verdicts (this rules out the path by which the format check first rejects the all-missing input as a format error and so masks the missing-data gate).

4. The tester feeds into the same verdict surfaces an input stating missing explicitly and an input stating a measured 0 explicitly and confirms whether the two verdicts are distinguished from each other, and, after recording missing in the measurement record layer, performs a store-and-retrieve round trip and confirms whether the retrieved value has changed to 0 or a default value.

5. If a separate display or aggregation layer that recomputes the verdict exists, the tester checks that layer's output and contribution to the total score for the same missing input, and observes whether the missing data is restored to a pass or full score.

6. To confirm the detection power of the probe (a standing checking mechanism that counts the verdict surfaces that fold missing data into a pass), the tester feeds in an input reproducing an absence-pass defect that actually occurred in the past and confirms whether the probe counts it as a violation, and, as a placebo control, also confirms that the probe counts no violation for a normal input.

Pass criteria

- Across all verdict surfaces of the population, the numbers of cases in which the all-missing input and the partially missing input returned a pass-family answer (pass,

'normal' or full score) are each 0, 0 verdict surfaces are left unchecked, and 0 items have no corresponding verdict surface in the population.
- The verdict for the missing input and the verdict for the measured-0 input are distinguished from each other, missing remains 'none' after the store-and-retrieve round trip, and missing data is not produced as a pass or full score in the recomputation of the display and aggregation layers or in the contribution to the total score.
- The probe counts the defect-reproducing input as a violation and counts no violation for the normal input, and the probe's standing execution point and most recent execution time are presented as records.

Record items

- Count and list of the verdict-surface population, excluded verdict surfaces and the reasons, and the result of the coverage comparison against items
- For each verdict surface (verdict for the all-missing input, verdict for the partially missing input, whether missing and measured 0 are distinguished)
- Values before and after the store-and-retrieve round trip, and the results of the display and aggregation layer recomputation and of the contribution to the total score
- Results of the probe self-verification (for the defect-reproducing input and the normal input, respectively), and the standing execution point and most recent execution time

> Caution — **pitfall**: An all-missing input is rejected first at the format check and readily yields 'not a pass', but actual missing data arrives with only one field empty, and the verdict surface then fills that place with 0 and awards full score; unless a partially missing input is fed in as well, this test passes while the absence-pass defect remains in place.

C.2.3 PAVS-MEA-03 — Confirms that, in a measurement input used for verdicts or correction, the state in which the stimulus changes but the value stays unchanged is handled not as a valid signal but as a measurement fault, and that the change-detection threshold is above the noise level.

Test type: injection

Preconditions

- The complete list of measurement channels used for verdicts or correction can be queried.
- Whether the stimulus condition, observation window and change-detection threshold of each channel are declared can be identified channel by channel.
- A copy in which the value is fixed to a constant at the output point of a measurement channel, and a copy containing an interval without stimulus, can be executed on the production path.

Procedure

1. The tester enumerates every measurement channel used for verdicts or correction, fixes them as the population, and counts the channels for which the stimulus condition, observation window and change-detection threshold are declared and the channels for which they are not declared.

2. The tester sets the injection target as the whole population if it has 10 or fewer entries or, if it has more than 10, as 10 channels — the 5 with the most dependent items and 5 drawn at random from the rest — and records the draw result.

3. For each target channel, the tester performs a normal test, records as the baseline the range of variation in the declared stimulus interval and the dispersion (standard deviation) in

the no-stimulus interval, and compares the declared change-detection threshold with that dispersion.

4. The tester executes a copy in which the stimulus is left unchanged and the value is fixed to a constant only at the output point of the measurement channel, and observes whether the system indicates that channel as a measurement fault, whether the reason is stated as liveness not met, and what the verdicts of the dependent items are.

5. If the channel is an input to closed-loop correction, the tester also observes the correction output and confirms whether the injection of correction continues on the belief that the unchanging input is a valid error, or whether a correction stop or a test abort is left in the record.

6. The tester executes a placebo copy in which the same channel is unchanging in an interval without stimulus and confirms whether a misfire occurs, and compares, across the whole of the output data, the number of channels recorded as constant series with the number of channels to which a liveness predicate is wired.

Pass criteria

- In the population, there are 0 channels for which the stimulus condition, observation window or change-detection threshold is undeclared, 0 channels to which no liveness predicate is wired, and 0 channels recorded as constant series without a wired predicate.
- In every unchanging-signal injection copy, the channel concerned is indicated as a measurement fault with liveness not met as the reason, the dependent items are not produced as a pass (not-measured or fail), and closed-loop correction does not continue to use that value, with a correction stop or a test abort left in the record.
- The declared change-detection threshold is at least 3 times the dispersion observed in the no-stimulus interval, and there are 0 misfires of the fault indication in the placebo copy.

Record items

- Count and list of the channel population, whether each channel has the declarations, and the draw method and list of the injection targets
- 3-way comparison table, for each channel, of the range of variation in the stimulus interval, the dispersion in the no-stimulus interval and the declared threshold
- Fault indication reasons, dependent-item verdicts and closed-loop correction handling results of the injection copies
- Number of misfires in the placebo copy, and the number of channels with a wired liveness predicate against the number of constant-series channels

> Caution — **pitfall**: If a perfect constant is injected, the fault is caught however low the threshold is, so the test passes; but an actually faulty channel carries least-significant-digit noise along with it, and if the threshold is set lower than the dispersion of that noise, the noise is counted as ‘change’ and the predicate in principle never fires in operation.

C.2.4 PAVS-MEA-04 — Confirms that the soundness of measurement channels is proven channel by channel not by a self-consistency cross-check but by the arrival of a known reference stimulus at the end point, and that, when there is no stimulus, its marker does not appear at the end point.

Test type: demonstration

Preconditions

- The complete list of measurement channels used for verdicts can be queried, and a mutually distinguishable reference stimulus (a unique marker value or a known physical target) can be defined for each channel.
- A test with the reference stimulus injected can be executed on the production path, and the end-point outputs (measurement records and results documents) can be inspected.
- At least 1 copy reproducing a measurement defect that actually occurred in the past can be constructed.

Procedure

1. The tester enumerates every measurement channel used for verdicts and assigns a marker value to each channel such that the ‘expected end-point values’, computed by first applying the unit conversion, rounding and significant digits of the scoring path, do not overlap between channels.

2. The tester performs, on the production path, a test with the reference stimulus injected into each channel, and checks whether the observed values read from the end-point outputs agree with the expected end-point values within the declared tolerance.

3. As the no-stimulus control, the tester performs through the same path a test into which no reference stimulus at all is injected, and confirms whether there is any channel whose marker value appears in the end-point outputs.

4. The tester traces backward into which value of which item each injected stimulus was carried and builds a channel→end-point-field correspondence table, keeps a list of the channels for which the backward trace failed, and also checks the verdicts of the items that depend on those channels.

5. To confirm the detection power of the probe (a checking mechanism that determines channel soundness by arrival at the end point), the tester feeds in the measurement-defect reproduction copy (unwired channel, mismatched coordinate conventions or the like) and observes whether the probe fires a fail.

6. The tester confirms whether any check remains that claims soundness solely by reading back configured values inside a channel and, if one remains, confirms whether its result is cited in the outputs as grounds for soundness.

Pass criteria

- For every channel used for verdicts, the injected marker is found within the tolerance of the expected end-point value or, for each non-arriving channel, the items that depend on that channel are handled as not-measured (0 cases in which the marker did not arrive and yet a dependent item was produced as a pass).
- There are 0 channels whose marker value appears in the end-point outputs of the no-stimulus control run.
- The probe fires a fail on the measurement-defect reproduction copy, and there are 0 items for which a configured-value read-back cross-check is cited alone as the grounds for the soundness determination.

Record items

- Correspondence table of channel–marker value–expected end-point value–observed value–tolerance
- Number of channels successfully traced backward, the list of non-arriving channels and the verdicts of the items that depend on those channels
- Results of the no-stimulus control run (number of channels in which the marker appeared), and the firing result and reason for the defect reproduction copy
- Probe execution time and test configuration identifier

> Caution — **pitfall**: The observation that the marker ‘was found’ holds even if the value in fact came from another channel or was a constant hard-coded in the end-point field from the start,

> so without advance computation of the expected end-point values and a no-stimulus control, an unwired channel is recorded as a sound channel; and a criterion that is passed merely by ‘disclosing’ the non-arriving channels passes even a system in which most channels never reach the end point.

C.2.5 PAVS-MEA-05 — Confirms that a proxy metric that is a function of numerical-analysis conditions declares its comparable range and conditions in machine-readable form, that those conditions are recorded with each output result, and that attempts at out-of-range comparison are actually blocked while comparisons under the same conditions are not blocked.

Test type: injection

Preconditions

- The complete list of metrics used in scoring, and a field indicating whether a metric is a function of numerical-analysis conditions (integration step, resolution, mesh and the like), can be queried.
- Two or more results of the same metric produced under different numerical-analysis conditions can be prepared.
- Recently issued results documents can be inspected.

Procedure

1. The tester enumerates every metric used in scoring and counts the presence and fill rate of the machine-readable field indicating whether a metric is a function of numerical-analysis conditions, and, if 0 metrics are classified as proxies, records that fact as ‘no such metric’ and does not record it as a pass.

2. The tester determines the sample of proxy metrics — all of them if there are 20 or fewer, and, if there are more than 20, 20 metrics consisting of the 10 most frequently cited in recently issued results documents and 10 drawn at random from the rest — and records the draw time and the list drawn.

3. For each metric in the sample, the tester confirms whether the comparable range and conditions are declared not as sentences in a document but as machine-readable fields, and whether, for each result produced, the numerical-analysis conditions that produced that result are recorded in the result itself.

4. For each metric in the sample, the tester feeds in an input attempting an absolute-value comparison of two results produced under different numerical-analysis conditions and an input attempting a direct comparison with a real-environment measurement value, and observes whether the computation is refused, whether the reason for blocking is recorded, and whether the compared value is kept out of the results document.

5. As a placebo control, the tester attempts a comparison of two results produced under the same numerical-analysis conditions and confirms whether it is produced normally (this avoids mistaking for ‘blocking’ a state in which every attempt fails because the comparison function itself does not exist).

6. The tester draws 10 recently issued results documents at random (all of them, if fewer than 10 have been issued) and counts the items in which an absolute threshold is attached to a proxy metric without a condition fingerprint.

Pass criteria

- For every proxy metric in the sample, the comparable range and conditions are declared as machine-readable fields, and the numerical-analysis conditions are recorded in each output result of that metric.

- Every attempt at absolute-value comparison between results under different conditions and every attempt at direct comparison with real-environment values is blocked with the reason recorded, the compared values do not appear in the results document, and the placebo control attempting a comparison under the same conditions is produced normally.
- In the sample of results documents, there are 0 items in which an absolute threshold is attached to a proxy metric without a condition fingerprint.

Record items

- Total number of metrics, number classified as proxies and fill rate of the classification field (if there are 0 proxies, recorded as ‘no such metric’)
- Sampling method, draw time and list of sampled metrics
- Result of each comparison-blocking attempt (blocked/passed through, and the reason) and whether the placebo control was produced normally
- Number of results documents sampled and number of items with an absolute threshold attached without a condition fingerprint

> Caution — **pitfall**: A system in which no comparison holds at all — because it has no cross-condition comparison function or does not record numerical-analysis conditions in its results — passes because every blocking attempt appears ‘blocked’, but in operation a violation occurs the moment the figures of two results documents are copied side by side; without the placebo control of a same-condition comparison and the confirmation of per-result condition records, this test records the absence of the qualification mechanism as the operation of the qualification mechanism.

C.2.6 PAVS-MEA-06 — Confirms that, for each newly created or changed measurement channel, whether valid values are recorded in actual output data has been checked, that all-0, all-null and constant series are treated not as normal but as measurement-defect candidates, and that channels not registered in the history do not slip outside the audit.

Test type: audit

Preconditions

- The history of the creation and change of measurement channels is recorded together with the time of each change.
- The test output data at the time of the preceding audit and at a recent time can each be opened and the value column of each channel inspected directly.
- A copy in which a particular channel produces all 0 can be constructed.

Procedure

1. The tester enumerates from the history every measurement channel newly created or changed since the preceding audit (the most recent 12 months, if there is no audit history).

2. The tester compares the channel sets of the outputs at the time of the preceding audit and of the recent outputs, enumerates as ‘omitted from history’ the channels that exist only in the recent outputs and are not registered in the history, and combines them with the preceding list to fix the population.

3. The tester takes the whole population as the sample if it has 20 or fewer entries or, if it has more than 20, draws 20 at random as the sample, and records the draw method, random seed and sample list.

4. For each sampled channel, the tester opens the output data of the first actual test after the creation or change and inspects the value column directly, classifies it as one of valid values, missing, all 0, all null or constant series according to the rule that a column is

classified as a constant series if it has 1 distinct value after rounding to the declared significant digits, and checks whether a ‘first actual output confirmation’ record (confirmer, confirmation time, identifier of the output confirmed, and determination) exists.

5. The tester obtains the number of channels that are all 0, all null or constant series across the whole of the output data by counting the value columns directly, not from the output of anomaly detection, and compares it with the number of defect-candidate registrations, and, for each ‘reason why the constant is normal’ given for an unregistered channel, confirms whether that reason designates the identifier of an evidence output showing that the constant is the only value that can occur in the interval concerned.

6. The tester selects 1 channel from the sample, executes a copy that produces all 0 and observes whether it is classified as a defect candidate, and confirms, as a placebo control, that a normally varying channel is not misclassified as a defect candidate.

Pass criteria

- There are 0 channels omitted from history, and for every channel in the sample a first actual output confirmation record exists, with the output confirmed designated by its identifier.
- Every all-0, all-null and constant-series channel is registered as a defect candidate or, for each unregistered channel, a ‘reason why the constant is normal’ designating the identifier of an evidence output is stated channel by channel (a reason without an identifier is counted as not met).
- The all-0 injection copy is classified as a defect candidate, and the normally varying channel of the placebo control is not misclassified as a defect candidate.

Record items

- Number of channels registered in the history, number of channels omitted from history and population count, and the sampling method, random seed and sample list
- Value-column classification result for each sample (including the significant digits used for the constant-series determination) and whether a first actual output confirmation record exists
- Comparison of the number of all-0, all-null and constant-series channels with the number of defect-candidate registrations, and whether the reasons for non-registration have evidence identifiers
- Classification results of the all-0 injection copy and of the placebo control

> Caution — **pitfall**: If the audit population is taken to be only ‘the channels registered in the creation and change history’, a channel added without leaving a history remains outside the audit and keeps flowing all 0; and if the constant-series determination has no significant-digit rule, a faulty column is classified as ‘varying’ on the strength of least-significant-digit noise alone, and the defect hardens into normal.

C.2.7 PAVS-MEA-07 — Confirms that, for every test condition that affects scoring, cases in which the submitted value is not actually applied are detected, that the requested value, the effective value and whether it was applied are stated side by side in machine-readable form in the result outputs, and that the effective value is a read-back of the result of application.

Test type: injection

Preconditions

- The complete list of test condition entries that affect scoring, and whether a means of confirming the effective value is declared for each entry, can be queried.

- A copy in which the value of a condition is submitted but only its application path is disabled can be executed on the production path.
- The fields of test result outputs can be queried by machine reading.

Procedure

1. The tester fixes the population not as ‘entries for which a means of confirming the effective value is declared’ but as ‘every test condition entry that affects scoring’, records its count, and keeps a list of the entries among them for which no confirmation means is declared.

2. The tester performs 1 normal test on the production path, counts whether all three fields — requested value, effective value and whether applied — are carried for each entry in the result outputs as machine-readable fields, and keeps a list of the entries with omissions, counting as omitted any entry that remains only in the execution log and is absent from the result outputs.

3. For the entries for which no confirmation means is declared, the tester confirms whether the verdicts of the items that depend on such an entry are handled as not-measured or produced as a pass.

4. The tester sets the injection target as the whole population if it has 5 or fewer entries or, if it has more than 5, as 5 entries — the 3 with the most dependent items and 2 drawn at random from the rest — and, for each target, executes on the production path a copy in which the value is submitted but only the application path is disabled, so that the effective value remains the default value.

5. In each copy, the tester observes whether mismatch detection fires and in which form the firing is handled — test abort, fail or an explicit mismatch indication — and executes a placebo copy in which the requested value and the effective value actually agree to confirm whether a misfire occurs.

6. The tester follows the production path of the effective value and confirms whether what is read back is the value applied to the physics and execution, or a re-read of the same configuration store that holds the submitted value.

Pass criteria

- For every entry in the population, the requested value, the effective value and whether applied are stated side by side in the result outputs as machine-readable fields, with 0 omissions, and there are 0 items that depend on an entry with no declared confirmation means and are produced as a pass.
- In every application-disabled copy, the mismatch is detected and the result is not produced as a pass, and there are 0 detection misfires in the placebo copy.
- For every injection target, it is confirmed through the production path that the means of confirming the effective value is a read-back of the result of application, not a re-query of the configuration store.

Record items

- List and count of the condition-entry population, and the list of entries with no declared confirmation means
- Number of entries in the normal run that satisfy the side-by-side statement of the three fields, and the list of entries with omissions (including entries left only in the log)
- Draw method of the injection targets, the form of detection firing and handling result for each copy, and the result of the placebo copy
- Result of confirming the production path of the effective value for each entry (read-back of application / re-query of configuration)

> Caution — **pitfall**: If the population is taken to be ‘entries for which a confirmation means is declared’, conditions that declare no confirmation means drop structurally out of the audit

> target, and omissions of the side-by-side statement and silent non-application never come to light; and if what is read back is the same store that holds the submitted value, the requested value and the effective value always agree, so that detection in principle cannot fire, and yet the test passes.

C.2.8 PAVS-MEA-08 — Confirms that the performance limits of the SUT derive not from constants of the scoring system but from the manifest, that raising and lowering changes to the manifest are reflected in measured behavior in both directions, and that, for an SUT that declares no limit, a system constant does not silently become the upper bound.

Test type: injection

Preconditions

- The SUT manifest declares performance limits (speed, torque, joint limits and the like), and the same scenario can be re-executed with only the manifest changed.
- A measurement channel that measures the behavior corresponding to the target limit exists, and the provenance of that channel is declared as ‘measured’ and passes the backward trace of production paths in **PAVS-MEA-01**.
- A copy of the manifest that declares no limit, and 1 SUT that declares no limit, can be prepared.

Procedure

1. The tester enumerates the limit entries used in scoring and confirms, not from declarations but from execution output records, whether the effective value of each entry came from the manifest or from a system default value.

2. The tester performs 1 normal run and confirms whether each entry is actually binding in that scenario (whether the measured behavior reaches that limit), and selects the injection targets from among the entries confirmed to be binding.

3. The tester confirms through the production path that the measurement channel to be used for observation is of measured provenance and not of the commanded-value family (because a command-family record always moves along when the command is changed, nothing is proven without this confirmation).

4. The tester executes a copy in which the manifest limit is lowered (for example, halved) and a copy in which it is raised (to a value exceeding the value suspected to be a system constant), and observes whether the measured behavior moves in both directions and whether the measured value of the raised copy stops at the same value as in the original run.

5. The tester executes a copy with that limit declaration removed from the manifest, and observes whether the system rejects the test or handles it as not-measured, rather than silently substituting a system constant, and the reason.

6. The tester executes a configuration in which a condition narrower than the manifest limit is enforced on the test side and confirms whether the fact of enforcement and the enforced value are stated in the result, and performs the same procedure at least once for each kind of test (lane), including at least 1 SUT that declares no limit.

Pass criteria

- In both the lowered and the raised copies, the measured behavior moves following the change so that the measured values of the three runs (original, lowered and raised) are distinguishable from one another, and there are 0 entries for which the measured value of the raised copy stops at the same value as in the original run.

- In the copy with the limit declaration removed, the test is rejected or handled as not-measured, and the number of cases scored with a system default value applied instead is 0.
- The result of the run with an enforced condition states the fact of enforcement and the enforced value, and in the checks for each kind of test and the check of the SUT that declares no limit, the number of cases in which a system constant acted as the upper bound is 0.

Record items

- List of limit entries, the result of confirming by execution the origin of each entry (manifest/default value), and the result of the binding check
- Result of confirming the provenance of the observation channel (measured / command-family)
- Measured values of the three runs (original, lowered and raised), and the handling result and reason for the copy with the limit declaration removed
- Number of runs performed and results by kind of test and by SUT (including the SUT that declares no limit)

> Caution — **pitfall**: If the limit is changed only downward, the measured value appears to follow the manifest because of the property that ‘the smaller one wins’, but at higher values the system constant is still the upper bound, so the cap stays hidden; and if the behavior is observed through a commanded-value-family channel, only the record moves along while the physics stays the same, so a state in which ‘the specifications changed but the physics stayed the same’ goes through as a pass.

C.3 VDT — Verdict test methods (8)

C.3.1 PAVS-VDT-01 — Confirms whether an item for which the required measurement does not hold is not counted as a pass but is produced as a third state (not-measured) — or, for a mandatory item, as a fail with the reason ‘absence of measurement’ — and whether the not-measured state is preserved through to the aggregate and the results document.

Test type: injection

Preconditions

- The scoring engine produces item-level verdicts and an aggregate for the test as a whole, and the verdict of record is preserved in machine-readable form.
- The verdict of record of 1 test and its input measurement records can be fed back into the re-scoring path, and a results document and the aggregate display surface can be generated from that re-scoring output.
- As the control, 1 normal test result containing at least 1 passing item and at least 1 failing item is available, and the threshold direction of each item (whether the item passes when the value is at or above the threshold or when it is at or below the threshold) is stated in the item definition.

Procedure

1. The tester feeds the measurement records of the control into the re-scoring path once, as a copy in which nothing has been changed, and confirms whether all item verdicts and the 4 aggregate values agree in every case with the original verdict of record (re-scoring determinism placebo control). If they do not agree, that test is not used as the control for this test.

2. The tester selects 2 items into which missing data is to be injected — 1 item that passes when the measurement value is at or above the threshold, and the other 1 an item that

passes when the measurement value is at or below the threshold, so that a value of 0 is a pass. The tester also records **whether each item is declared a mandatory item**. The tester records the pre-injection verdicts of the two items and the 4 aggregate values (pass count, fail count, not-measured count and pass rate — for an implementation that does not compute a pass rate, that fact).

3. The tester makes a copy in which only the samples of the measurement channel required by each item are replaced with missing (absent) — the item definitions, thresholds and seeds are not changed — re-scores it, and observes the verdict state values of the two items and the 4 aggregate values.

4. The tester compares the verdicts of all remaining items into which missing data was not injected with their pre-injection state, and confirms that the scoring engine has not collapsed the whole test into not-measured.

5. The tester recomputes the pass-rate formula from the figures in the output, generates a results document and the aggregate display surface from the same re-scoring output, and confirms whether, on those surfaces too, the not-measured count is indicated as a separate state distinguished from pass and fail.

6. The tester additionally re-scores a copy in which the required measurements of all items are replaced with missing, and observes what value the pass rate is indicated as under the condition in which the denominator becomes 0.

Pass criteria

- Neither item is counted as a pass — in particular, the item in the direction in which a value of 0 is a pass is not counted as a pass. The verdict state is indicated as not-measured or, for a mandatory item, as a fail whose reason is identified as ‘absence of measurement’. If an item that is not a mandatory item is counted as a fail, the result is nonconforming.
- For an implementation that computes a pass rate, its denominator equals the sum of the pass count and the fail count, the recomputed value equals the value in the output, and in the all-items-missing copy the pass rate is not produced as a figure such as 100% or 0% but is indicated as not computable. For an implementation that does not compute a pass rate, the not-measured and absence-of-measurement counts in the all-items-missing copy are preserved in the aggregate as separate states. In either case, not-measured is indicated as a separate state in the results document and on the aggregate display surface.
- The verdicts of the non-injected items are identical in every case to those before injection, and the verdicts and aggregate of the placebo-control copy are also identical in every case to the original.

Record items

- Identifiers, measurement channels and form of absence representation of the 2 injected items, and the threshold direction of each item
- Item verdict states and the 4 aggregate values before and after injection, and the pass-rate indication of the all-items-missing copy
- Numbers of agreements and disagreements for the placebo control (copy without injection) and the collapse control (non-injected items)
- Whether not-measured is indicated in the results document and on the aggregate display surface, and the scoring code version, seed and configuration identification information used for re-scoring

> Caution — **pitfall**: A scoring engine that replaces missing data with a value of 0 happens to produce a fail depending on the threshold direction and so satisfies ‘not counted as a pass’; unless an item in the direction in which a value of 0 is a pass is injected as well, a scoring engine that cannot distinguish missing from a value of 0 passes this test. Also, an

> implementation that, on the strength of the allowance of an ‘absence-of-measurement fail’ for mandatory items, folds **all** missing data into fails loses the not-measured state — unless it is also observed whether injection into an item that is not mandatory produces not-measured, and whether an absence-of-measurement fail is distinguished by reason from ‘a fail that arose because the value violated the threshold’, an implementation whose 3-valued verdict is in effect 2-valued passes.

C.3.2 PAVS-VDT-02 — Confirms whether a pass is produced only on the positive evidence that every required condition is met by measurement, and whether the default-value path, the unmet-condition path and the non-measured-provenance path return no pass.

Test type: injection

Preconditions

- A list of the verdict output surfaces that can indicate a pass — the verdict of record, score reports, conformance indications, aggregates, display layers and the like — is declared in a document.
- The item definition enumerates every required condition for a pass, and each measurement value carries its measurement provenance (measured/commanded value/default value/estimate).
- Measurement records and event records can be composed at will and fed into each verdict output surface.

Procedure

1. The tester obtains the declared list of verdict output surfaces, extracts every path that produced a pass indication from the issue history of the most recent 12 months (all issues, if fewer than 100 were issued in that period) and compares them with the list. If a path not on the list is found, the tester adds that path to the test target and records that fact.

2. The tester enumerates every required condition for a pass from the item definition of the test-target item and builds a condition roster.

3. The tester feeds an input whose measurement records are entirely empty (empty item set, 0 samples) into every output surface in the test target, and counts the pass indications in the values produced.

4. For each condition in the condition roster, the tester composes a boundary input in which only that 1 condition is unmet and the rest are met by measurement, feeds in as many such inputs as there are entries in the roster, and records the verdict value of the item for each round.

5. The tester feeds in an input in which every required condition has a satisfying value but the measurement provenance of 1 of them is not measured (derived from a default value or a commanded value) and observes the verdict, and then feeds in an input in which every condition is met by measurement and confirms whether a pass is produced (placebo control — this rules out the passing of logic that cannot produce a pass).

6. The tester feeds in an input containing only a value by which the test scenario declared its own success, and confirms that this value alone cannot produce a pass.

Pass criteria

- For the empty input, there are 0 pass indications across all verdict output surfaces in the test target, and each surface produces a value meaning absence of a verdict (not-measured, not-checked or the like).

- Among the boundary-input rounds for all conditions in the condition roster, the rounds in which a pass is produced number 0 (the number of test rounds equals the number of entries in the condition roster).
- No pass is produced for the satisfying input whose measurement provenance is not measured or for the self-declaration-only input, and a pass is produced for the input in which every required condition is met by measurement.

Record items

- List of test-target verdict output surfaces, scope of the issue-history comparison (period and count) and any additionally discovered paths
- Required-condition roster and the verdict for each boundary-input round (pairs of condition identifier and verdict value)
- Output values for the empty input and the count of pass indications, and the verdicts for the non-measured-provenance satisfying input, the self-declaration-only input and the placebo-control input
- Input composition of each feed and the scoring code version at the time of injection

> Caution — **pitfall**: If only 1 arbitrary required condition is left unmet, a condition that the verdict logic never refers to in the first place — a condition declared in the item definition but omitted from the computation formula — goes untested; unless every condition in the condition roster is left unmet in turn, one at a time, logic that ignores some of the conditions passes this test.

C.3.3 PAVS-VDT-03 — Confirms whether the success metrics are conditioned not on the issuing of commands but on measured physical events, so that a no-op of the execution subsystem is not counted as a success.

Test type: injection

Preconditions

- The item definition declares the correspondence between every declared success metric and the physical event that each metric presupposes (change of state, contact or movement of the object).
- A test configuration that makes the execution subsystem a no-op (absence of the drive mechanism, drive commands ignored or the like) exists, and the fact of that configuration is noted in the test record but is not included in the input to the scoring engine.
- As the control, 1 normal successful test with the same scenario and the same seed is available.

Procedure

1. The tester performs the control test in the normal configuration and records the values of all declared success metrics together with the corresponding physical-evidence values (object displacement, contact force, video and trajectory).

2. For each physical-evidence channel used for cross-validation, the tester checks the measurement provenance to determine whether its value is measured or a value derived from an execution command (forced attachment, direct pose update or the like), and excludes command-derived channels from the basis for cross-validation.

3. The tester queries the entire input received by the scoring engine to confirm whether it contains a field from which the no-op configuration can be identified and, if there is one, also performs scoring on a copy with that field removed.

4. With the same scenario and the same seed, the tester performs the test in a configuration in which only the execution subsystem is made a no-op, observes in the verdict of record the

values of all success metrics and the physical-evidence values that each metric referred to, and confirms the displacement of the object numerically from the output video and trajectory records.

5. The tester compares the physical-evidence values of the control and the no-op test to confirm whether the two conditions yield distinguishable values, and confirms that the no-op test reached the production of per-item verdicts rather than a crash or an indefinite wait.

Pass criteria

- In the no-op configuration, none of the declared success metrics is counted, and the no-op test terminates normally with the verdict of the item produced as fail or not-measured.
- The physical-evidence values on which success counting was based are recorded together in the verdict of record, the measurement provenance of those channels is measured, and the object displacement in the no-op test is less than 1/10 of the threshold of the item, whereas the displacement in the control exceeds the threshold.
- The result is the same on the copy with the no-op identification field removed, and the normal control is counted as a success under the same procedure.

Record items

- Method of making the subsystem a no-op, the location where that fact is recorded, and the result of the query on whether it is exposed in the scoring input
- Pairs of the values of all success metrics and the physical-evidence values for both tests (control and no-op), and the object displacement figures and the threshold of the item
- Measurement provenance classification for each physical-evidence channel (measured/commanded value/default value/estimate) and the channels excluded from cross-validation
- Result of confirming that the seeds and configurations of the two tests match, termination states, and identification information of the video and trajectory outputs

> Caution — **pitfall**: If the no-op configuration is exposed in the scoring input as a flag, the scoring engine treats the test as a failure on the basis of that flag rather than physical evidence, so the test passes while an actual no-op in the field (failure of the drive mechanism) is still counted as a success; unless it is confirmed that the copy with the flag removed gives the same result, test-specific exception handling is mistaken for physical cross-validation.

C.3.4 PAVS-VDT-04 — Confirms whether, at the verdict layer, a measurement failure is not compensated for with substitute values, default values or mock values but is explicitly reported together with a reason that distinguishes the kind of failure.

Test type: injection

Preconditions

- The list of measurement channels used for scoring and the failure representation of each channel (absence of samples, channel exception and impossible-value notation) are declared.
- Each value field of the output carries its measurement provenance (measured/commanded value/default value/estimate).
- The permitted scope of mock data is declared in a document, and a result field indicating the fact and scope of its use exists.

Procedure

1. The tester selects 1 channel required for the verdict, injects each of 4 measurement-failure conditions individually — all samples absent, channel exception, impossible value

(out-of-range notation, negative distance or the like) and only some samples absent — and scores each, and also performs 1 normal test with nothing injected in the same configuration.

2. In the verdict of record for each injection condition, the tester observes the state value of the item and the recorded reason.

3. The tester queries every value field that the item refers to, making the determination not by the magnitude of the value but by the measurement provenance indication of each field, and flags any field whose provenance is a default value or an estimate but for which no failure reason is recorded alongside.

4. The tester compares the value fields of each failure condition with the same fields of the normal test, down to the displayed digits, and confirms whether any field produces an identical value under the failure condition as well (detection of constants and of reuse of the preceding test's values).

5. The tester compares the reasons for the four conditions with one another, and confirms that absence, exception, impossible value and partially missing data are not folded into the same handling but are reported distinctly from one another.

6. The tester performs 1 each of a test into which mock data is deliberately fed and a normal test, and confirms whether the indication field is produced as ‘used (scope included)’ and ‘not used’, respectively.

Pass criteria

- Under none of the four failure conditions is the item produced as a pass; all samples absent, channel exception and impossible value are reported as failure or not-measured together with a reason, and partially missing data is reported either with the effective sample count stated alongside or as not-measured.
- There are 0 value fields whose measurement provenance is a default value or an estimate but which have no failure reason, and 0 cases in which a value field under a failure condition is identical to the value of the normal test down to the digits.
- The four conditions are reported with mutually distinguishable reasons, and the test with mock data fed in indicates the fact and scope of use, while the normal test indicates no use.

Record items

- Injected channel, definitions of the four failure conditions and configuration of the normal test
- Verdict state and verbatim reason string for each condition, and a comparison table of whether the reasons are distinguishable
- Number of fields covered by the exhaustive value-field query, number of fields flagged on the basis of measurement provenance, and number of identical values flagged against the values of the normal test
- Values of the mock-data indication field (for the mock-input test and the normal test, respectively)

> Caution — **pitfall**: If substitute values are searched for only by the magnitude of the value (0 or a constant), a measured 0 cannot be distinguished from a substituted 0, and an implementation that fills failures with 0 is reported as having ‘0 substitute values’; unless the measurement provenance indication and an identity comparison with the values of the normal test are performed as well, silent substitution passes this test.

C.3.5 PAVS-VDT-05 — Confirms whether non-physical output that bypasses the physics simulation path is not accepted as scoring evidence, and whether that determination is based not on self-declaration but on measured characteristics of the physical output.

Test type: injection

Preconditions

- Whether a test was performed on the physics path is left in each test record in machine-readable form.
- An item that discriminates non-physical output (kinematic motion, forced attachment and teleportation) is declared, and its scope of application is stated in a document.
- A complete list of the kinds of test subject to scoring exists as a document.

Procedure

1. The tester compares the declared scope of application of the discriminating item with the complete list of kinds of test subject to scoring and enumerates every uncovered kind. If there are 3 or fewer uncovered kinds, the tester puts all of them into the test target; if there are 4 or more, the tester draws 3 with a recorded random seed and puts those into the test target.

2. The tester checks the input of the discriminating item to determine whether it is the physics-path self-declaration field of the test record or measured characteristics of the physical output (contact force, falling acceleration, penetration depth and trajectory continuity).

3. In a kind of test for which the discriminating item is declared, the tester performs a test in a configuration with the physics path switched off (non-physical motion or forced attachment), and also performs a normal physical control with the same seed.

4. In a copy of the output of the non-physical test, the tester changes only the field recording whether the physics path was performed to ‘physical’, leaves the measurement content unchanged, re-scores the copy, and observes whether the discriminating item still fires.

5. In the verdicts of record of the three tests (non-physical, forged-self-declaration copy and normal physical), the tester observes whether the discriminating item fires, the overall verdict and the evidence-not-accepted indication field, and confirms in what state the success metrics of the non-physical test appear in the aggregate and the results document.

6. The tester injects the same non-physical configuration into every uncovered kind of test drawn in step 1, and confirms whether each result is counted as a pass.

Pass criteria

- The discriminating item fires in both the non-physical-configuration test and the forged-self-declaration copy, and does not fire in the normal physical control.
- The result of the non-physical-configuration test is not counted as evidence for a pass, and the non-acceptance of evidence is explicitly indicated in the verdict of record and the results document.
- There are 0 kinds of test for which the discriminating item is not declared, or the non-physical test results for every uncovered kind drawn in step 1 are treated as not-measured and the list of uncovered kinds is made public.

Record items

- Complete list of kinds of test subject to scoring, scope of application of the discriminating item, list of uncovered kinds, and the draw seed and number drawn
- Kind of input of the discriminating item (self-declaration field/measured characteristics) and the re-scoring result of the forged-self-declaration copy

- Summary of the verdicts of record of the three tests, value of the evidence-not-accepted indication field, and actual field value of the record of whether the physics path was performed
- Injection results for the uncovered kinds of test and the counting status of each result

> Caution — **pitfall**: If the discriminating item reads only the self-declaration field for whether the physics path was used, it does not fire on non-physical output whose field is marked ‘physical’; if injection is performed only with honestly marked copies, a circular verification is formed in which the self-declaration and the verdict support each other, and a discriminator that does not look at measured characteristics passes.

C.3.6 PAVS-VDT-06 — Confirms whether, in the declared strictness order, a more lenient layer is unable to change a fail or warning produced by a stricter layer into a pass, either by overwriting or by subsequent registration.

Test type: injection

Preconditions

- The verdict layers and their strictness order are declared in a document, and the verdict recording location of each layer is identified.
- A test path exists through which a write to the verdict of record can be attempted via the same interface as a normal writer.
- The change history of the verdict of record (recording entity, time and previous value) is preserved.

Procedure

1. The tester obtains the declared order document, extracts every actual recording entity from the change history of the verdict of record for the most recent 12 months (all, if there were fewer than 100 verdicts in that period) and compares them with the declared list. If an entity not on the list is found, the tester adds that entity to the test target.

2. The tester selects 3 tests for which the strict layer produced a fail, a warning and not-measured, respectively, and for each injects an attempt, as a writer of the lenient layer, to record a pass for the same test (for the warning case, a pass without the warning).

3. The tester observes the return values of the three injected attempts, the final value and the change history of the verdict of record, and the notification channel in which the rejection reason is left.

4. If there is a path through which the lenient layer registers a new verdict for the same test identifier under the label of re-scoring or retest, the tester also attempts to register a pass through that path, and observes whether the earlier strict verdict is preserved in the history and which verdict the final indication shows.

5. After the four attempts, the tester newly generates the results documents and aggregates of those tests, and confirms that the display and aggregation layers do not recompute the verdict with their own formulas and restore it to a pass.

6. The tester performs 1 normal write by the strict layer itself and confirms whether it succeeds (over-blocking check).

Pass criteria

- All four attempts are rejected or, even where registration is allowed, the final indication remains the stricter verdict; the value of the verdict of record is identical to that before the attempts, and the earlier strict verdict is preserved in the change history.

- The results documents and aggregates newly generated after the injection indicate the same verdicts as the verdict of record (for the warning case, the warning indication is retained).
- The rejection reason is notified through a channel that leaves a record, and the normal write by the strict layer succeeds.

Record items

- Declared order, scope of the history comparison (period and count), list of actual recording entities, and any additionally discovered entities
- For the four attempts (overwriting of a fail, a warning and not-measured, and registration under the label of re-scoring), the return values, rejection reasons, and whether the verdict-of-record value and the history changed
- Verdict indications in the results documents and aggregates generated after the injection
- Result of the over-blocking check (normal write by the strict layer) and the time of each attempt

> Caution — **pitfall**: If the block is attached only to the operation called ‘overwrite’, the lenient layer can add a new pass to the same test under the label of re-scoring or retest and make it read as the latest value; unless the history preservation of the earlier strict verdict and the final indication are also observed, the order appears to be upheld.

C.3.7 PAVS-VDT-07 — Confirms whether, when a defect in the verdict criteria has been corrected, all existing verdicts of the defect period are retrospectively audited against the new criteria, verdict shifts and recipient notifications are recorded, and the audit population has not been narrowed by self-classification.

Test type: audit

Preconditions

- The change history of the verdict criteria (item definitions, thresholds and verdict logic) is preserved together with the time, scope and classification of each change.
- The correspondence between issued score reports and their recipients is recorded in a ledger.
- The measurement records of past tests are preserved so that they can be re-scored against the current criteria.

Procedure

1. The tester enumerates the entire change history of the most recent 12 months, taking as the audit population not the range operator's ‘defect correction’ classification but every change that touched item definitions, thresholds or verdict logic. The tester sorts the roster in ascending order of change time and draws the sample with a recorded random seed — from changes classified as defect corrections, 10 if there are more than 10 and all if there are 10 or fewer, and from changes classified as not a defect, an additional 5 (all, if there are fewer than 5).

2. For each sampled change classified as not a defect, the tester checks the basis of that classification — confirming whether evidence that there is no verdict shift (a re-scoring result, or a record showing that the scope of the change does not reach verdict production) exists.

3. For each sampled defect correction, the tester checks in the audit record the definition of the defect period (time of defect introduction and time of correction) and the size of the audit population, and compares that size with the number of verdicts issued in the same period as recorded in the ledger.

4. In the audit result for each sampled defect correction, the tester checks the counts of verdict shifts by direction (pass→fail, fail→pass and shift to not-measured) and the statement of the reasons for the shifts, checks for every score report in which a shift occurred the evidence of recipient notification (notification time and confirmation of dispatch or receipt), and compares the interval between the audit completion time and the notification time with the notification deadline declared by the range operator.

5. For each sampled defect correction, the tester selects at random 2 verdicts of the defect period — 1 that the audit record lists as ‘shifted’ and 1 that it lists as ‘not shifted’ — re-scores them against the current criteria, and cross-checks whether the results agree with the verdicts in the audit record.

Pass criteria

- A retrospective audit record exists for every sampled defect correction, and the size of the audit population agrees with the number of verdicts issued in the same period in the ledger (where they disagree, the audit record contains a list of the omitted cases and the reasons).
- For every sampled defect correction, verdict shifts are stated as counts by direction, and for each score report in which a shift occurred there is evidence of recipient notification within the declared notification deadline.
- The verdicts of all re-scoring cross-checks (2 per sample) agree with the verdicts in the audit record, and evidence of no verdict shift exists for every sampled change classified as not a defect.

Record items

- Definition and size of the population, sample size by classification, and draw method (sort key and random seed)
- For each sample, the defect period, audit population size, result of the ledger comparison and number of omissions
- For each sample, counts of verdict shifts by direction, form of the notification evidence, and interval between the audit completion time and the notification time
- For each re-scoring cross-check, whether its verdict agrees with the verdict in the audit record, and the result of checking the basis of the samples classified as not a defect

> Caution — **pitfall**: If the population of the retrospective audit is set by the auditee's self-classification as ‘defect correction’, merely labeling the changes that moved verdicts as improvements or feature changes reduces the audit target to 0; unless changes classified as not a defect are included in the sample and the basis of their classification is cross-checked, this audit holds only on paper.

C.3.8 PAVS-VDT-08 — Confirms whether a test directive requiring a pose or condition outside the SUT manifest is rejected without silent approximation, and whether the rejection reason and the limit value actually enforced are reported.

Test type: demonstration

Preconditions

- The manifest of the SUT declares joint and actuation limits, rated payload, and workspace and geometric reach.
- Every location at which the same limit is declared is identified, and the values at each location can be compared.
- Test intake and execution can be performed through the normal operating path, intake responses and execution records are preserved, and a time limit for the intake response is declared.

Procedure

1. The tester enumerates every location at which the same limit is declared, compares the values at each location and, if the values disagree, records both the loosest value and the strictest value.

2. The tester submits through the normal intake path 3 tests, 1 in each of 3 limit categories — joint and actuation limits, rated payload, and workspace and geometric reach — each containing a directive that clearly exceeds the declared limit.

3. The tester observes the intake response and execution record of each test, classifies the outcome as one of three — rejection at intake, explicit abort after execution, or scored after approximation — and confirms whether the response arrived within the declared response time limit.

4. In the record of each test, the tester checks that the requested value and the actually applied value are stated side by side to determine that the directive was not executed while clamped to the limit, and, if there is any interval during execution in which the applied value differs from the requested value, confirms whether that fact is recorded and reported.

5. The tester checks the content of the rejection reason (the limit entry violated, the requested value and the permitted range) and the notification channel in which the reason is left, and determines which of the per-location values compared in step 1 corresponds to the permitted range stated in the reason.

6. The tester changes the same directives to values within the limits, submits all 3, and confirms whether they are executed and scored normally (over-blocking check), and confirms whether the rejected tests are not counted in the pass/fail aggregate but are recorded in a rejected state.

Pass criteria

- The limit-exceeding tests in all three categories end in rejection or explicit abort, a response is given within the declared response time limit, and no verdict is produced as a pass.
- The requested value and the actually applied value are stated side by side, there are 0 tests scored while the two differ, and the rejection reason includes the limit entry violated, the requested value and the permitted range, and is recorded and notified.
- All 3 within-limit control tests are executed and scored normally, the rejected tests are not counted in the pass/fail aggregate, and the permitted range used for the rejection agrees with the value declared in the manifest (it is not the loosest of the duplicate declarations).

Record items

- Complete list of the locations at which the limits are declared, the value at each location, and the number of value disagreements
- Requested values submitted for each category, intake response, termination state and response time, and the record stating the requested value and the actually applied value side by side
- Verbatim rejection reason string, the location where it is recorded and notified, and the permitted range stated in the reason
- Results of the 3 within-limit control tests and the aggregate status of the rejected tests

> Caution — **pitfall**: If the same limit is also declared in duplicate at locations other than the manifest, only the loosest value is actually enforced and a directive that cannot be executed is accepted as ‘within limits’; if only one degree of freedom is tested and the values are not compared across the declaration locations, a limit that is not enforced appears to be upheld.

C.4 VRF — Verification test methods (16)

C.4.1 PAVS-VRF-01 — Confirms that a negative control produces a fail through the firing of its designated item, that the verdict actually divides on the two sides of the threshold (violating and non-violating), and that the automatic comparison with the expected verdict fires when a defect is injected.

Test type: injection

Preconditions

- For every representative item and core item of each profile, a negative control and its paired positive control are registered in the regression suite, and each control declares its expected verdict, the identifier of its target item and its deviation width from the threshold of that item.
- The scoring engine version and the scenario corpus version are fixed and are not changed during the test.
- A record channel exists through which the results of a suite run can be viewed at item level, and the tester has the authority to make test injections and to restore them.

Procedure

1. The tester obtains the list of controls and records the number of controls, the target item identifiers and the threshold deviation widths, and lists separately the representative items that have no corresponding negative control or paired positive control and the controls whose deviation width exceeds a pre-declared multiple so that they cannot test the boundary.

2. The tester executes the suite once in the unchanged state and observes the actual verdict of each negative control and the identifier of the item that produced its fail, and the actual verdict of each paired positive control.

3. The tester injects a temporary change of the declared expected verdict of 1 negative control to ‘pass’, re-executes the suite, and observes whether the automatic comparison flags the disagreement.

4. The tester injects a defect that makes the item designated by that control always return a pass regardless of conditions, re-executes the suite, and observes whether the negative control is flagged as a disagreement and the suite as a whole is reported as failed.

5. The tester injects a defect that makes the same item always return a fail regardless of conditions, re-executes the suite, and observes whether the paired positive control is flagged as a disagreement and the suite as a whole is reported as failed.

6. The tester restores the 3 injections, re-executes the suite and confirms the return to the state of step 2, and then records the content, injection time and restoration time of each injection.

Pass criteria

- In the unchanged run, the verdict of every negative control is a fail and the item to which that fail is attributed is identical to the declared designated item, and the verdict of every paired positive control is a pass.
- In each of the 3 injections (tampering with the expected verdict, always pass and always fail), the suite flags the control concerned as a disagreement and reports the overall result as failed.
- The sum of the number of representative items without a corresponding control and the number of controls that cannot test the boundary is 0, or the item concerned is indicated as ‘boundary unverified’ and that indication appears in the results document of any verdict that includes that item.

Record items

- List of controls and, for each control, the target item, expected verdict and threshold deviation width
- Actual verdict of each control in the unchanged run and the item to which each fail is attributed
- Content, injection time, result reported by the suite and result of the restoration check for each of the 3 injections
- Identifiers of the scoring engine version and the scenario corpus version

> Caution — **pitfall**: The path by which an over-strict item that always returns a fail regardless of conditions lets every negative control pass — without a paired positive control, only ‘it catches violations’ is confirmed and ‘it lets non-violations pass’ is never confirmed even once, so a suite that has verified a constant rather than a threshold stays permanently green.

C.4.2 PAVS-VRF-02 — Confirms that, when the scoring system is changed, re-scoring of the golden set detects and counts verdict shifts by direction (including the direction to not-measured), and that those shifts are not erased by re-baselining the golden set and are disclosed in the operational record.

Test type: injection

Preconditions

- A golden set whose verdicts have been settled and frozen exists together with its item-level verdicts, and each entry is accompanied by its seed, configuration and scoring code version, and by grounds independent of the output of the scoring system (the original observation records).
- The entire golden set can be re-scored from the same measurement input, and whether the re-scoring path is a re-evaluation of the measurements or a re-performance of the execution is distinguished in a document.
- Test changes can be injected and restored, and the version change history of the golden set and the shift-disclosure entries of the operational record can be viewed.

Procedure

1. The tester records the number of golden-set entries and the verdict distribution (pass/fail/not-measured), and obtains the number of entries excluded from re-scoring and the list of reasons for exclusion.

2. The tester re-scores the entire golden set from the same measurement input in the unchanged state, and observes the number of verdict differences by entry and by item.

3. The tester injects a shift of the verdict threshold of 1 item toward the measured distribution of that item so that at least 1 verdict changes, and then re-scores the entire set.

4. The tester puts the measurement channel referred to by 1 item into the missing state so that the verdict of that item shifts to not-measured, re-scores the entire set, and observes whether that shift is counted in the shift count and whether the missing data is not folded into a pass or a fail.

5. The tester compares the shift counts and directions reported for the two injections with the values computed by directly comparing the original verdicts with the new verdicts.

6. The tester restores the injections, and confirms whether the fact of the shifts has been registered in the operational record together with a statement of their justification, and whether the version history of the golden set contains an update of expected verdicts made to fit these shifts.

Pass criteria

- The item-level verdict differences in the unchanged re-scoring number 0, and the re-scoring population is the same as the entire golden set (0 exclusions).
- In each of the two injections, the shift count and direction (pass↔fail and shift to not-measured) are reported, and those values agree with the tester's independently computed values.
- The fact of the shifts and their justification are registered in the operational record and can be viewed, and the history contains 0 instances of a shift being absorbed into an update of the golden set's expected verdicts without an approval record.

Record items

- Number of golden-set entries, verdict distribution, and number of excluded entries and the reasons
- Number of differences in the unchanged re-scoring and the classification of the re-scoring path used
- Content of the 2 injections and the shift counts and directions (values computed by the system and values computed independently by the tester)
- Registration identifier and registration time in the operational record, and the version change history of the golden set

> Caution — **pitfall**: The re-baselining path, by which, once a shift is detected, the expected verdicts of the golden set are overwritten with the new output so that the differences are 0 from the next re-scoring onward — unless the version history is compared with the approval records, the golden set becomes not ‘a memory of the verdicts before the change’ but a mirror of the current scoring system, and any regression, once it has passed a single time, remains normal permanently.

C.4.3 PAVS-VRF-03 — Confirms whether a comparative-test report confirmed the effective value of the independent variable individually in each test group, whether the effective values of the two groups actually differed beyond the measurement resolution, and whether a comparison in which they did not was classified not as a refutation but as unverified.

Test type: audit

Preconditions

- The complete list of comparative-test reports for the period under audit (the most recent 12 months, or the period since the most recent revision) has been settled and frozen, and reports whose conclusion is ‘no difference’ are also included in the list.
- The auditor can access the original measurement records and the execution configuration records of each report.
- The resolution and uncertainty of the measurement channel that carries the independent variable are declared.

Procedure

1. The auditor settles and freezes the population N of comparative-test reports for the target period by comparing it with the verdict ledger, then draws at random a sample of size n = min(N, max(5, ⌈√N⌉)) and records the random seed used. All reports cited in external documents as the basis for a refutation or an improvement are included in the sample.

2. For each report in the sample, the auditor compares the declared independent variable with the existence of effective-value confirmation records for group A and group B respectively — observation location, observation time and observed value — and with their source (whether execution output data or a read-back of the configuration).

3. The auditor compares the difference between the effective values of the two groups with the declared resolution and uncertainty, and determines whether the intended treatment was actually applied.

4. The auditor selects 1 report from the sample, directly recomputes the effective values of the independent variable for the two groups from the original measurement records, and compares them with the reported values.

5. The auditor identifies the reports in which the effective value was confirmed in only one group, for which there is no confirmation record, or in which the effective difference between the two groups is within the resolution, and confirms whether the conclusions of those reports are classified as unverified.

Pass criteria

- In every report in the sample, the effective value of the independent variable is recorded as an observed value for each of group A and group B, and that record derives from execution output data and not from a read-back of the configured value.
- In every report in the sample, the difference between the effective values of the two groups exceeds the declared resolution and uncertainty, and the auditor's recomputed value agrees with the reported value within the tolerance declared by the item concerned.
- Every report with an unconfirmed effective value or an unconfirmed effective difference is classified as unverified and has not been cited as the basis for a refutation or improvement claim.

Record items

- Population size, sample size, random seed and identifiers of the sampled reports
- For each sample, the independent variable, the effective value for each group (observation location, time and value) and the classification of its source
- Result of comparing the effective difference between the two groups with the declared resolution and uncertainty
- Recorded value and auditor-computed value for the 1 recomputation comparison, and the number of reports with an unconfirmed effective difference and their current classification status

> Caution — **pitfall**: The path by which the effective values of the two groups are each recorded but are equal to each other within the measurement resolution, so that a comparison in which the treatment was never once applied passes as ‘effective value confirmed’ — if only the existence of the values is counted and the difference between the groups is not, an experiment that in effect had no independent variable remains standing as a refutation.

C.4.4 PAVS-VRF-04 — Confirms that, in cases where a test failure was attributed to a defect of the SUT, a refutation test of the measurement chain using a traceable known stimulus preceded the attribution, and that the refutation is valid for the configuration of the chain at the time of attribution.

Test type: audit

Preconditions

- The complete list of cases attributed to an ‘SUT defect’ in the target period has been settled and frozen.
- The auditor can view the investigation records and raw measurement data of each case and the configuration change history of the measurement channel concerned (calibration constants, path and version).

- The procedure of the measurement-soundness refutation test — the independent basis for the true value of the known stimulus, the injection point, the end-point observation location, the pass condition and the validity period — is specified in a document.

Procedure

1. The auditor freezes the population N of cases attributed to an SUT defect, draws n = min(N, max(5, ⌈√N⌉)) at random and records the random seed. All attribution cases reflected in issued results documents are included in the sample.

2. For each sampled case, the auditor compares the time of the attribution decision with the time at which the measurement-soundness refutation test was performed, and confirms whether the refutation preceded the attribution.

3. The auditor checks against the change history whether there was a configuration change to the channel concerned between the time the refutation was performed and the time of the attribution decision, and confirms whether the specified validity period was exceeded.

4. The auditor confirms whether the true value of the known stimulus derives from a basis independent of the channel under test (a reference instrument or an independent observation), and checks against the chain description whether the stimulus path was one that passed through to the end point of the measurement channel in question.

5. The auditor re-performs the refutation test of 1 sampled case by the same procedure and compares the observed value with the recorded value.

Pass criteria

- In every sampled case, the time the refutation was performed precedes the time of the attribution decision, there was no configuration change to the channel concerned in between, and the interval between the refutation and the attribution decision is within the specified validity period.
- The refutation record contains the injection of a known stimulus whose true value was fixed on an independent basis, together with the end-point observed value, and has not been replaced by a read-back of configured values internal to the channel or by a confirmation that ‘a value is being output’.
- The observed value of the auditor's re-performance agrees with the recorded value within the declared tolerance.

Record items

- Population size, sample size, random seed and identifiers of the sampled cases
- For each sample, the time of the attribution decision, the time the refutation was performed, and the configuration change history in between
- Basis for the true value of the known stimulus used in the refutation, injection point, end-point observation location and observed value
- Conditions and result of the 1 re-performance by the auditor

> Caution — **pitfall**: The path by which a refutation that took the true value from the channel's own output passes — a faulty measurement that returns values matching the directive always passes a check that takes itself as the reference, and even a refutation that was once valid guarantees nothing at the time of attribution if the calibration constants have changed in the meantime.

C.4.5 PAVS-VRF-05 — Confirms whether, when the specification of record is declared in multiple locations, tampering at each location is flagged by the authenticity comparison, whether the location that is physically enforced is identified and recorded, and whether a silent clamp of a directive outside the specification is not produced as a normal verdict.

Test type: injection

Preconditions

- At least 1 SUT model to be compared and the corresponding manufacturer-confirmed specification documents are available.
- A list of the locations at which specification entries are declared (configuration declaration, model description, motion-planning input and the like) is documented.
- There is authority to change and restore a test copy of the model, and the original model is not changed.

Procedure

1. The tester compares the joint limits, actuation limits, geometry and material properties of the model copy with the manufacturer-confirmed specification entry by entry, and records the number of mismatched entries and the location at which each entry is declared.

2. For each declaration location in turn, once each, the tester injects a change of 1 joint limit, at that location only, to a value outside the manufacturer specification (exceeding the permitted range by 30%), executes the authenticity comparison check, and observes whether each location is flagged and whether the value mismatch between declaration locations is flagged as a separate entry.

3. In each injected state, the tester executes once a scenario in which that joint reaches the vicinity of its limit, compares the measured joint value with the value at each declaration location, and identifies the location that the physics engine actually enforced.

4. In the run with the injection at the enforced location, the tester observes whether, where the directive is cut back to the limit (clamped), that fact is reported as missing or as a fail, and whether a verdict is not silently produced.

5. The tester executes 1 test with a model on which the authenticity comparison has not been performed, and observes whether the indication ‘unverified model’ appears in the verdict produced and in the results document.

6. The tester confirms **whether two or more values for the same entry are found in the manufacturer's published materials** and, if they are, observes whether the adopted baseline and the reason for adopting it are declared in the comparison record.

7. The tester exhaustively enumerates the entries for which the model value differs from the baseline, and confirms, **for each entry, the existence of a basis record and the source to which that basis points**. The tester separately tallies the number of entries without a basis.

8. The tester deletes from the test copy the basis record of 1 difference entry whose basis is recorded, re-executes the comparison check, and observes whether that entry is newly flagged as a ‘difference without a basis’.

9. The tester restores all injections and confirms that the comparison check reports 0 mismatches, and then records the injection and restoration history and the identified enforced location.

Pass criteria

- The injections at all declaration locations are flagged in the authenticity comparison check, value mismatches between locations are flagged separately, and verdicts

produced with a flagged model do not reach an issuable state (test rejection or unverified indication).
- The measured enforced value agrees with the declared value at the identified enforced location, that location is recorded in a document as the enforced location, and there are 0 cases in which a normal verdict was produced in a run in which a directive clamp occurred.
- The fact of being unverified is indicated in verdicts and results documents produced with a model on which the comparison has not been performed.
- For entries with multiple manufacturer-published values, the adopted baseline and its reason are declared, a basis record exists for every entry that differs from the baseline, and the entry whose basis was deleted is flagged as a ‘difference without a basis’.
- In the reporting of differences, the **direction (model wider/narrower)** is indicated for each entry.

Record items

- Number of compared entries, mismatched entries and the declaration location of each entry
- Values before and after injection at each location, the flagging results of the comparison check and whether the mismatch between locations was flagged
- Measured enforced value observed in the run, value at each declaration location, and the form of reporting when a clamp occurred
- Document entries in which the unverified indication appeared, and the verdict identifiers
- List of entries with multiple manufacturer-published values, and the adopted baseline and reason
- All entries differing from the baseline (value, direction, presence or absence of a basis record and its source), and the number of entries without a basis
- Entry newly flagged in the basis-deletion injection

> Caution — **pitfall**: The path by which the injection is made only at a location that the physics engine does not read, so that the comparison check also passes as ‘modified’ while the behavior does not change at all, and the path by which the engine silently cuts a directive outside the specification back to the limit and lets the run complete normally — an authenticity pass in which only the numbers have changed and a clamped normal verdict both conceal the fact that a specification different from the manufacturer specification is being enforced. **Also, if the manufacturer's published materials are taken from only one place, the baseline is silently set to one side** — even if another material from the same manufacturer gives a different value for the same entry, that fact is not left in the comparison record, and a model fitted to a baseline set in that way passes as a ‘match’. The adoption of the baseline is itself part of the verdict.

C.4.6 PAVS-VRF-06 — Confirms whether a consistency claim is accompanied by the measurement method, sample and uncertainty, and whether the held-out sample was actually separated from the calibration sample along a condition axis and selected in advance, and was not contaminated by repeated viewing.

Test type: audit

Preconditions

- The complete list of consistency claims and calibration-value adoption cases in the target period has been settled and frozen.
- The conditions of the held-out test (the selection time of the held-out sample, the separating condition axis and the pass threshold) are specified in advance in a document.

- The calibration-value adoption history (version, adoption time, basis data and data-usage record) is preserved.

Procedure

1. The auditor freezes the population N of calibration-value adoption cases, draws n = min(N, max(5, ⌈√N⌉)) at random and records the random seed. All cases whose consistency claim is cited in external documents are included in the sample.

2. For each sampled case, the auditor confirms the presence of a statement of the measurement method, sample size and uncertainty, recalculates the uncertainty from the sample size and the stated formula, and compares it with the stated value.

3. The auditor checks against the data-usage record whether the selection time of the held-out sample precedes the computation time of the calibration value, and whether the held-out sample was not used as input to the calibration process.

4. The auditor checks against the lists of condition-axis values of the two samples whether the held-out sample was separated from the calibration sample so that their value ranges do not overlap on at least 1 condition axis.

5. The auditor compares the number of times the held-out test was performed on the adopted calibration-value version with the history of calibration-value modifications after the held-out result was observed, re-performs the held-out test of 1 sampled case according to the specified procedure, and compares the residual and the pass/fail outcome with the recorded values.

Pass criteria

- The consistency claims of all sampled cases state the measurement method, sample size and uncertainty, and the auditor's recalculation of the uncertainty agrees with the stated value.
- The selection of the held-out sample preceded the computation of the calibration value, the held-out sample was not used as calibration input, and its value range does not overlap with that of the calibration sample on at least 1 condition axis.
- The number of held-out test performances on the adopted version is 1 or, where it is more than one, there are 0 calibration-value modifications after the held-out result was observed, and the pass/fail outcome of the held-out test re-performed by the auditor is the same as recorded.

Record items

- Population size, sample size, random seed and identifiers of the sampled cases
- For each sample, the measurement method, sample size and uncertainty (stated value and recalculated value)
- Selection time of the held-out sample, computation and adoption times of the calibration value, and number of held-out test performances
- Condition-axis value ranges of the held-out sample and the calibration sample, and the conditions, residual and pass/fail outcome of the re-performed held-out test

> Caution — **pitfall**: The path by which a sample split at random under the same conditions as the calibration sample is used as the held-out sample, or the calibration value is modified until it passes the held-out test and only the last pass is kept — unless the condition-axis separation and the number of performances are counted, the held-out test becomes not a test of generalization but part of the calibration loss, and collapses at the first real application in which the conditions change.

C.4.7 PAVS-VRF-07 — Confirms whether calibration prioritized the identification of omitted physics over increasing the number of parameters, and whether the final adoption was justified not by the training residual but by the held-out residual.

Test type: audit

Preconditions

- The calibration history is recorded per calibration round (estimated parameter set, number of degrees of freedom, training residual, held-out residual and held-out test result).
- The composition and size of the training sample and the held-out sample of each round, and the number of condition axes, are recorded.
- The complete list of calibration events in the target period has been settled and frozen.

Procedure

1. The auditor freezes the population N of calibration events, draws n = min(N, max(3, ⌈√N⌉)) at random, and records the random seed used.

2. From the series of rounds of each sample, the auditor tabulates the chronological change in the number of degrees of freedom, the training residual and held-out residual of each round, and the training sample size and number of condition axes.

3. For each round in which the degrees of freedom increased, the auditor confirms the existence of a preceding omitted-physics review record, and checks whether that record contains a named physics term and an observational basis supporting the introduction or rejection of that term (the change in held-out residual upon introduction, or the measurement basis for the reason for rejection).

4. The auditor compares the degrees of freedom and held-out residual of the finally adopted round with the values of the round with the minimum held-out residual and of the round with the maximum degrees of freedom in the series, and checks the statement of the reason for adoption.

5. The auditor classifies, sentence by sentence, whether the statement of achieved consistency for each adoption case is based on the training residual or on the held-out residual.

Pass criteria

- A preceding omitted-physics review record exists for each round that increased the degrees of freedom, and that record contains the named physics term and the observational basis (0 cases consisting only of a re-confirmation of declaration documents).
- The finally adopted calibration value is accompanied by a held-out test pass record, and its held-out residual is the minimum in the series or, where it is larger, the reason for adoption is recorded.
- There are 0 adoption cases that state the achievement of consistency solely on the basis of a reduction in the training-sample residual.

Record items

- Population size, sample size and random seed
- For each sample, the change in degrees of freedom across the series of rounds, the training and held-out residuals, and the training sample size and number of condition axes
- Named physics terms in the omitted-physics review records, and the presence or absence of an observational basis
- Degrees of freedom and held-out residual of the finally adopted round, the values of the round with the minimum held-out residual in the series, and the reason for adoption

> Caution — **pitfall**: The path by which only the formal requirement is met by appending a single line, ‘omitted physics was reviewed’, each time the degrees of freedom are increased — unless the named physics terms and the observational basis for introduction or rejection are checked, the review record becomes an alibi for parameter expansion, and the fact that a low-degree-of-freedom round with a better held-out residual in the series was discarded without a reason does not come to light either.

C.4.8 PAVS-VRF-08 — Confirms whether the determination of a defect's cause was based on measurement made beforehand, whether corrections targeting the same component were stopped once 2 had failed and the matter was transferred to an independent entity, and whether that requirement was not evaded by renaming or splitting the component.

Test type: audit

Preconditions

- Investigation records are kept per hypothesis (hypothesis statement, the component named as the cause, basis measurement, content of the correction and retest result), and the auditor can view the actual change target of each correction.
- An independent review entity has been designated and documented in advance, and its intake channel leaves a record.
- The complete list of investigation cases in the target period has been settled and frozen.

Procedure

1. The auditor freezes the population N of investigation cases and draws n = min(N, max(5, ⌈√N⌉)) at random, but includes in the sample all cases with 3 or more correction attempts.

2. For each sampled case, the auditor compares the start time of the first correction with the creation time of the measurement record cited as its basis, and confirms whether the measurement preceded the correction.

3. The auditor views the actual change target of each correction, reclassifies the cause component by the auditor's own criteria, compares the result with the recorded classification, and recounts the number of correction attempts on the same component.

4. In the records following 2 failures as counted under the reclassification, the auditor confirms the stopping of corrections and the transfer notification (receiving entity and notification time).

5. For transferred cases, the auditor confirms whether the independent review entity is distinct from the entity that performed the corrections, and whether intake and review records remain and the review has been reflected as the basis for subsequent corrections.

Pass criteria

- In every sampled case, the basis measurement record of the first correction was created before the start time of the correction.
- Under the auditor's reclassification, the number of 3rd correction attempts on the same cause component is 0, or a transfer notification record exists from before the start of the 3rd attempt.
- For every transferred case, intake and review records exist from an independent review entity distinct from the entity that performed the corrections.

Record items

- Population size, sample size, random seed and identifiers of the sampled cases
- For each sample, the hypothesis statements, the recorded cause components, the auditor's reclassification results and the number of differences between them
- Creation time of the basis measurement and start time of each correction

- Identifiers and times of the transfer notification, intake and review records, and whether the review entity is distinct

> Caution — **pitfall**: The path by which, while the same spot is being corrected, the name of the cause component is split into sub-entries in the record so that the count becomes ‘1 attempt each’ — unless the auditor opens the actual change targets and reclassifies them, the transfer requirement is never triggered even once although the same place is corrected five times, and on the record each correction becomes a new hypothesis.

C.4.9 PAVS-VRF-09 — Confirms whether a case in which a defect was attributed to environmental or concurrency factors was preceded by establishment of the reproduction frequency, comparison with normal cases and isolated reproduction by standalone execution.

Test type: audit

Preconditions

- The complete list of incidents attributed to ‘environmental or concurrency factors’ in the target period is settled and frozen.
- The execution records of each incident (number of trials, number of occurrences, whether execution was concurrent, and the original execution records) are preserved, and the minimum number of trials required for a frequency determination is specified in advance in a document.
- Test resources with which the auditor can attempt reproduction by standalone execution are secured.

Procedure

1. The auditor freezes the population N of incidents attributed to environmental or concurrency factors, draws n = min(N, max(5, ⌈√N⌉)) at random and records the random seed.

2. The auditor checks the number of trials and the number of occurrences of each sample, and classifies a case whose number of trials is below the pre-specified minimum number of trials as ‘frequency not established’ and a case that occurred in all trials as ‘deterministic reproduction’.

3. In the attribution basis of each sample, the auditor confirms the existence of a record of comparison with the execution records of normal cases and of a record of isolated reproduction by standalone execution, and checks whether the times of the two records precede the attribution decision.

4. The auditor places the two records side by side and counts whether the symptoms cited as the basis for attribution also appear, at the same frequency, in the same segment of the execution records of normal cases.

5. The auditor attempts to reproduce 1 deterministic-reproduction case by standalone execution for the pre-specified minimum number of trials, and compares the observed occurrence rate with the recorded occurrence rate.

Pass criteria

- The attribution basis of every sampled case contains both a normal-case comparison record and a standalone isolation record, the two records precede the attribution decision, and the number of trials is at or above the pre-specified minimum number of trials, so that there are 0 cases with frequency not established.
- Among the symptoms cited as the basis for attribution, there are 0 entries that appear identically in the execution records of normal cases as well.

- The occurrence rate observed in the auditor's standalone reproduction agrees with the recorded occurrence rate within the pre-specified decision rule, and no case that occurred in all trials is attributed to environmental or concurrency factors.

Record items

- Population size, sample size, random seed and identifiers of the sampled incidents
- Number of trials and number of occurrences for each sample, and the result of the classification as frequency not established or deterministic reproduction
- Times of the normal-case comparison and standalone isolation records, and the time of the attribution decision
- Number of trials and number of occurrences in the auditor's standalone reproduction, and the list of compared symptoms

> Caution — **pitfall**: The path by which a case recorded as ‘seen intermittently’ on the strength of 1–2 trials is closed by putting it down to the environment without the frequency having been established — unless the minimum number of trials is fixed, deterministic reproduction cannot be distinguished from intermittent failure, so a code defect that reproduces every time is closed as ‘the load was high at the time’ and the investigation of the cause does not even begin.

C.4.10 PAVS-VRF-10 — Confirms whether a regression declaration was contrasted with baseline trials of the same configuration and was made only after being distinguished from intermittent failure by a pre-specified minimum number of repetitions.

Test type: audit

Preconditions

- The complete list of regression declarations in the target period is settled and frozen.
- Each declaration is accompanied by the list of post-change trials and the list of baseline (pre-change) trials, together with configuration identifiers, seed design, scenario version, scoring code version and verdicts.
- The minimum number of repetitions and the decision rule required for a regression determination are specified in advance in a document, and the time at which that specification was established can be confirmed.

Procedure

1. The auditor freezes the population N of regression declarations, draws n = min(N, max(5, ⌈√N⌉)) at random and records the random seed. All declarations that caused a suspension of testing or a hold on issuance are included in the sample.

2. The auditor confirms whether the time at which the document specifying the minimum number of repetitions was established precedes the time of the sampled declarations, and whether that number of repetitions has a basis of calculation with respect to the failure rate to be detected.

3. The auditor counts the number of baseline trials and the number of post-change trials of each sample separately, and compares them with the pre-specified minimum number of repetitions.

4. The auditor checks whether the configuration identifiers, scenario versions and scoring code versions of the two sets of trials are identical except for the object of the change, and whether the record states which design was used — fixed-seed or varied-seed.

5. The auditor retests 1 sample with the baseline configuration and with the post-change configuration, each for the pre-specified number of repetitions, and compares the two failure rates by the pre-specified decision rule.

Pass criteria

- A list of baseline trials exists for every sampled case, the numbers of baseline and post-change trials are each at or above the pre-specified minimum number of repetitions, and the establishment of the specification precedes the declaration.
- The configuration identifiers, scenario versions and scoring code versions of the two sets of trials are identical except for the object of the change, and the seed design is stated in the record.
- The auditor's retests of both sides reach the same conclusion as the record (regression or intermittent failure) under the pre-specified decision rule.

Record items

- Population size, sample size, random seed and identifiers of the sampled declarations
- Numbers of baseline and post-change trials for each sample, and the list of verdicts for each trial
- Result of the identity check of configuration identifiers, scenario versions and code versions, and the seed design
- Number of trials and number of failures in the retests of both sides, and the decision rule applied

> Caution — **pitfall**: The path that measures the failure rate by repeating only the post-change trials and takes as its baseline the recollection that ‘it worked before’ — unless a list of baseline trials of the same configuration is required, a test that was intermittent from the outset is confirmed as a regression, so that the investigation of the cause begins in a place unrelated to the actual change and issuance comes to a halt.

C.4.11 PAVS-VRF-11 — Confirms whether a change to scoring or verdict code is accompanied by a usage-site impact analysis and by a ‘run against this submission’ of the representative regression for each profile, whether there is no incorporation path that bypasses the gate, and whether existing verdicts produced in reliance on the changed code are identified as subject to re-evaluation.

Test type: injection

Preconditions

- A list of representative kinds of test (at least 1 per operational profile) is declared in a document, and the corresponding tests are registered in the regression suite.
- The change-submission path has a gate that requires impact analysis and regression execution, and the execution time of each test can be viewed.
- A test change can be submitted and reverted, and the list of existing verdicts can be queried.

Procedure

1. The tester selects 1 computation element shared by two or more profiles, independently enumerates its usage sites in advance, and then submits, for test purposes, a change that gives its return value a deviation large enough to flip a verdict.

2. The tester observes whether, at submission, the gate requires and produces a list of usage-site impacts, compares the produced list with the list the tester enumerated in advance, and records the set difference.

3. The tester runs the regression suite, confirms by execution time whether the declared representative tests for each profile were executed after the time of this submission, and observes whether the tests of the profiles reached by the deviation are reported as failing.

4. The tester attempts to submit the same change through an incorporation path that does not pass through the gate, and observes whether it is incorporated and, if it is, whether that fact is left in the record and marked as subject to re-evaluation.

5. Leaving the change in the adopted state, the tester observes whether existing verdicts produced in reliance on that code are enumerated as subject to re-evaluation, and records their count.

6. The tester reverts the change, confirms that the regression suite returns to its original state, and then records the conditions for clearing the re-evaluation marking.

Pass criteria

- The list of usage sites produced by the gate includes the whole of the tester's independently enumerated list (0 omissions), and there are 0 changes incorporated without passing through the gate or, where one is incorporated, that fact is left in the record and it is marked as subject to re-evaluation.
- At least 1 declared representative test per profile is executed after the time of this submission, and failures are reported in the profiles reached by the deviation.
- Existing verdicts produced in reliance on the changed code are enumerated in the list of verdicts subject to re-evaluation, and their count agrees with the query result.

Record items

- Injected computation element, magnitude of the deviation and time of submission
- Set difference between the list of usage sites produced by the gate and the tester's independently enumerated list
- Execution times and verdict results of the representative tests for each profile, and whether the gate-bypassing submission was incorporated
- Count of existing verdicts enumerated as subject to re-evaluation, and the conditions for clearing the marking

> Caution — **pitfall**: The path by which the regression suite reuses the passing results of an earlier run as they are, so that the representative tests of other profiles are marked ‘passed’ without having been executed, and changes that skipped the gate through an emergency incorporation path — unless execution times are compared with the submission time, tests that never ran even once are counted as coverage, and cross-profile defects come to light only after the results document has been issued.

C.4.12 PAVS-VRF-12 — Confirms whether each entry of a claim of supported scope or verification capability has evidence of a ‘pass’ in an end-to-end test on the current version, and whether unverified entries are indicated distinctly from verified entries.

Test type: audit

Preconditions

- A published support list (SUT types, scenario types, profiles and the like) exists as a document, and its entries can be counted.
- For each entry, the evidence level (L0–L3) and the identifier of the supporting test can be queried, and the execution record and verdict of the supporting test are preserved.
- The scoring code version and model version of the supporting test can be compared with the current versions.

Procedure

1. The auditor determines the total number of entries M in the published support list, draws n = min(M, max(8, ⌈√M⌉)) at random and records the random seed. All entries asserted as ‘supported’ in external documents are included in the sample.

2. The auditor queries the identifier of the supporting test of each sampled entry, and confirms from the execution record whether that test was an end-to-end execution on the production path (L2 or higher) and whether its verdict is a pass (rather than not-measured or fail).

3. The auditor checks whether the scoring code version and model version of the supporting test are identical to the current ones and, where they differ, confirms whether the entry is marked as subject to reconfirmation.

4. The auditor identifies the entries whose basis is limited to registration, configuration, code existence or successful initialization (L0), and confirms whether those entries are indicated in the list distinctly as unverified.

5. The auditor re-executes the supporting test of 1 sample and compares the verdict produced with the recorded verdict.

Pass criteria

- Every sampled entry indicated as ‘verified’ has an end-to-end execution record at L2 or higher, and its verdict is a pass (0 entries based on not-measured or fail).
- Every entry whose supporting test differs from the current version in scoring code version or model version is marked as subject to reconfirmation.
- Every entry with only L0 evidence is indicated distinctly as unverified, and the verdict of the auditor's 1 rerun agrees with the recorded verdict.

Record items

- Number of entries M in the support list, sample size and random seed
- For each sampled entry, the evidence level, the supporting test identifier and the verdict of the supporting test
- Result of comparing the scoring code version and model version of the supporting tests with the current versions
- Number of entries with L0 evidence and whether they are indicated as unverified, and the conditions and produced verdict of the 1 rerun

> Caution — **pitfall**: The path by which a run that did go end to end but whose verdict was fail or not-measured is counted as ‘an L2 execution record exists’ — unless the existence of a run is distinguished from its passing, ‘we did try running it’ is promoted to ‘verified’, and even entries that passed only on an old version remain in the basis set of current capability claims.

C.4.13 PAVS-VRF-13 — Confirms that safety-related guards and stop mechanisms fire repeatedly under a known stimulus and bring about the declared subsequent state, and that they are not mishandled under placebo conditions, missing input or guard-coexistence conditions.

Test type: injection

Preconditions

- The list of guards to be verified and, for each guard, the firing condition, the expected behavior on firing (stop, deceleration, rejection or the like), the upper limit on firing latency and the required state after firing are declared in a document, and the numbers of repetitions of the firing trials and the placebo trials are declared in advance.
- Test conditions under which injection stimuli can be applied (simulation or an isolated real environment) and an independent means of interruption are secured.
- The firing result of a guard can be observed not through the guard's own report but through an independent channel (drive directives, measured behavior, video and the like).

Procedure

1. For each target guard, the tester injects a stimulus exceeding the declared firing condition for the number of repetitions declared in advance (at least 3), and observes in each trial, through the independent channel, whether the expected behavior occurs and the time from the stimulus to the onset of the behavior.

2. The tester applies stimuli below the firing threshold (placebo condition) as many times as declared in advance, a number at or above the number of firing trials, and counts the number of firings.

3. The tester puts one of the inputs referred to by the guard into a missing state, re-injects the firing-condition stimulus, and observes whether the guard passes silently or reports a failure or a firing.

4. The tester performs 1 injection of a condition in which two or more guards hold simultaneously, and observes through the independent channel whether the firings of the guards do not neutralize one another and all declared behaviors are realized.

5. In each trial in which firing was observed, the tester observes the state of the SUT after firing (whether it has stopped, whether its posture is maintained, whether there are dropped objects) and confirms whether the firing satisfies the declared subsequent state.

6. The tester records the injection history and whether the means of interruption operated, and restores the test conditions to their original state.

Pass criteria

- Every target guard exhibits, as observed through the independent channel, the declared expected behavior in all trials of the firing-condition stimulus, the firing latency is within the declared upper limit in all trials, and the post-firing state satisfies the declared subsequent state.
- The number of firings in the placebo-condition trials is 0, and in the re-injection under the missing-input state the guard does not pass silently but reports a failure or a firing.
- Under the guard-coexistence condition, no guard cancels the declared behavior of another guard.

Record items

- For each guard, the magnitude, number of trials and injection times of the injected stimulus, and the test conditions used
- For each trial, the values observed on the independent channel, the firing latency and the state of the SUT after firing
- Number of trials and number of firings under the placebo condition
- Handling results of the missing-input injection and the guard-coexistence injection, whether the means of interruption operated, and confirmation of restoration to the original state

> Caution — **pitfall**: The path by which a firing observed in only one trial is counted as ‘works’, and the path by which the firing signal was issued but driving in fact continues — without repeated trials and observation through an independent channel, a guard that operates only intermittently and a guard that has never once operated are both counted as ‘normal’ in every test, and the fact that the guard was skipped by an exception while its referenced input was empty is left nowhere in the record.

C.4.14 PAVS-VRF-14 — Confirms that any issued verdict is actually recomputed and reproduced by a single command, and that the seed, configuration and scoring code version are recorded and fixed in the result in step with the execution.

Test type: demonstration

Preconditions

- The verdicts to be reproduced have already been issued, and their identifiers can be queried.
- The usage of the reproduction command is documented, and the tester executes it in a working environment different from that of the original producer.
- The reproduction run does not change the original verdict record, and the tester can isolate the stored copy of the original outputs in a temporarily unreadable state.

Procedure

1. From the population of recently issued verdicts, the tester draws at random 3 with mutually different profiles, records the random seed and the draw ledger, and records, from each verdict record, the presence and values of the seed, configuration identifier and scoring code version fields.

2. For each case, the tester executes the documented single command with only the verdict identifier as its argument, and observes the number of times additional input or manual operation is required during execution.

3. The tester reproduces 1 of the 3 cases while the stored copy of the original verdict's outputs is isolated in an unreadable state, and observes whether the command completes by recomputation rather than by reading back the stored copy.

4. The tester compares the reproduced verdict with the original verdict item by item, compares numerical differences with the tolerance declared by the item concerned, and records whether the reproduction path is a re-evaluation of measurements or a re-execution, and whether the seed was fixed.

5. The tester confirms whether the generation time of the intermediate measurements in the reproduction outputs coincides with the time of the reproduction run.

6. For 1 case, the tester changes the seed value to a different value and re-executes, and observes whether the seed record in the outputs changes to follow the seed actually used, so that it is distinguishable from the original record.

Pass criteria

- All 3 cases are completed by a single command executed with only the verdict identifier, with 0 additional manual inputs, and the 1 case for which the stored copy of outputs was isolated is also completed by recomputation.
- The reproduced verdicts are identical to the original verdicts item by item, numerical differences are within the declared tolerances, and the intermediate measurements of the reproduction outputs were newly generated at the time of the reproduction run.
- The changed seed is recorded in the outputs of the rerun with the changed seed, distinguishing them from the original record.

Record items

- Identifiers and profiles of the 3 target verdicts, the random seed for the draw, and the values of the seed, configuration and code version fields of each case
- For the reproduction runs, the form of the command, environment identifier, elapsed time and number of requests for additional input, and the classification of the reproduction path
- Item-by-item comparison table of the reproduced verdicts and the original verdicts
- Result of the reproduction with the stored copy isolated, generation time of the intermediate measurements, and the recorded value of the seed-changed rerun

> Caution — **pitfall**: The path by which the reproduction command does not perform scoring again but reads back the existing verdict record or the stored copy of outputs and outputs an ‘identical verdict’ — since the generation time can be updated to the time of copying, it cannot distinguish read-back from recomputation; unless the stored copy is isolated, the reproducibility

> requirement is satisfied by copying files, and the fact that the seed and version fields have become unrelated to the actual run also does not come to light.

C.4.15 PAVS-VRF-15 — Confirms whether, after parameter adjustments to the same component have failed 2 times, it was re-examined by observation whether the design assumptions of that component hold under the current test conditions.

Test type: audit

Preconditions

- The parameter adjustment history is recorded per component (adjusted parameter, value and retest result), and the auditor can view the original retest verdict of each adjustment.
- The design assumptions (premises of application) of each component are declared in a document.
- The complete adjustment history for the target period is settled and frozen.

Procedure

1. Looking directly at the original retest verdicts across the complete adjustment history, the auditor independently reclassifies each adjustment as a success or a failure, freezes the population N of ‘2 or more failures on the same component’ incidents, draws n = min(N, max(5, ⌈√N⌉)) at random and records the random seed.

2. For each sample, the auditor confirms whether the records after the 2nd failure contain a design-assumption re-examination entry — the statement of the premise examined, whether it holds under the current test conditions, and the data on which the judgment is based — and compares its time with the time at which the 3rd adjustment was started.

3. The auditor checks whether the judgment in the re-examination as to whether the premise holds presents measurement or execution observations as its basis rather than a reconfirmation of the declaration document.

4. For cases judged as the premise not holding, the auditor confirms whether follow-up actions (structural change, restriction of the scope of application, re-evaluation of the results produced under that premise) were recorded.

5. In 1 sample, the auditor selects one declared design assumption, directly confirms by observation whether it holds under the current test conditions, and compares this with the recorded judgment.

Pass criteria

- On the basis of the auditor's reclassification, every sampled case has a design-assumption re-examination record made after the 2nd failure, and the time of that record precedes the start of the 3rd adjustment.
- The judgment in the re-examination as to whether the premise holds presents measurement or execution observation data as its basis, and there are 0 cases described only by a reconfirmation of the declaration document.
- Every case judged as the premise not holding has a record of follow-up actions, and the auditor's direct observation agrees with the recorded judgment.

Record items

- Population size, sample size, random seed and identifiers of the sampled incidents, and the number of cases in which the auditor's reclassification differs from the recorded classification
- For each sample, the component, adjustment series, number of failures and time at which the 3rd adjustment was started
- Statement of the premise, judgment and supporting data in the re-examination record

- Conditions and result of the auditor's 1 direct observation

> Caution — **pitfall**: The path by which a failed adjustment is written up as a ‘partial improvement’ or ‘condition check’, left out of the failure count and so does not trigger the re-examination requirement — unless the auditor looks directly at the retest verdicts and reclassifies them, the population itself is shrunk by the self-classification of the party under investigation, and even when adjustments built on an already broken premise continue to a 3rd round, the audit passes with ‘no relevant incidents’.

C.4.16 PAVS-VRF-16 — Confirms whether the self-checking mechanisms are actually being executed automatically, and whether their scope of execution covers the verdict surfaces that the mechanisms protect.

Test type: audit

Preconditions

- The list of self-checking mechanisms held by the scoring system is documented.
- The verdict surfaces (items, measurement channels and issuing surfaces) protected by each mechanism are declared.
- The entity and interval of automatic execution are defined.

Procedure

1. The tester collects the list of self-checking mechanisms and the declarations of the verdict surfaces protected by each mechanism.

2. For each mechanism, the tester queries the execution history of a recent period, and distinguishes from the records whether each run was an automatic run (periodic or at the time of a change) or a run initiated by a person.

3. For mechanisms with an automatic execution history, the tester compares the scope actually targeted by those runs with the protected verdict surfaces collected in the first step.

4. The tester lists the verdict surfaces outside the scope of execution, and confirms whether that list is stated as excluded scope in the scoring system's report.

5. The tester selects one mechanism arbitrarily, injects within its protected verdict surfaces a defect that the mechanism is meant to catch, and observes whether it fires in the next automatic run not initiated by a person.

Pass criteria

- For every self-checking mechanism, an automatic execution history exists at the set interval or at the time of the most recent change, and the records confirm that those runs were automatic.
- Where the scope of execution does not cover the whole of the protected verdict surfaces, the excluded scope is stated in the scoring system's report.
- For the injected defect, the self-checking mechanism fires in a run carried out without human intervention.

Record items

- List of self-checking mechanisms and the declaration of the protected verdict surfaces of each mechanism
- Execution history for each mechanism — executing entity (automatic/manual), date and time, and target scope
- List of verdict surfaces outside the scope of execution, and whether they are reported
- Stimulus, whether firing occurred, and time of firing in the injection test

> Caution — **pitfall**: If a person runs a mechanism manually to create an ‘execution history’, this test is passed, but the requirement of the clause (automatic execution) is not met. In a system

> in which records do not distinguish whether the executing entity is automatic or a person, this test cannot be performed, and in that case the result is ‘not-measured’.

C.5 REP — Reporting test methods (15)

C.5.1 PAVS-REP-01 — Confirms that claims of the function and supported scope of the scoring system carry an evidence-level field, that the level is checked against the referenced execution records, and that a ‘verification pass’ is not issued on evidence below L2.

Test type: injection

Preconditions

- The claims of the function and supported scope of the scoring system are maintained as a machine-readable list in which each entry has an evidence-level field and a reference to the execution record that grounds that level.
- The list of issuing entry points can be derived separately from each of two sources: the interface specification and the issuing-path records of the recent issuing ledger.
- The test can be performed on an isolated copy of the same version as the production issuing build.

Procedure

1. The tester derives the list of issuing entry points separately from each of the two sources, compares the two lists, and fixes their union — including the entry points that appear in only one of them — as the complete target set of this test.

2. The tester selects 2 entries marked ‘verification pass’ in the supported-scope list, taken from different function types, creates a test input in which only the evidence-level field is lowered from L2 to L1 while the substance of the evidence is left unchanged, and feeds it into every issuing entry point in the complete target set.

3. For each entry point, the tester observes whether issuing is rejected or whether the indication of the entry concerned is demoted from ‘verification pass’ to ‘unverified’.

4. The tester feeds a missing-data input, in which the evidence-level field of the same entry has been deleted, into the same paths and, if it is not rejected, opens the stored and issued result to confirm what value remains in that field — this determines whether the missing value is filled in with the highest level or with an arbitrary default value.

5. The tester feeds in a forged input in which the evidence-level field of an entry whose referenced execution records point only to L1-level records is raised to L2, and observes whether the issuing path checks the declared level against the referenced records and rejects the input.

6. The tester feeds the original input into each entry point once and confirms that it is issued normally (placebo control), and compiles into a comparison table the per-round input differences, whether issuing took place, the rejection reason strings, and the versions of the isolated copy and of the production issuing build.

Pass criteria

- The input with the lowered level and the input with the level forged upward are rejected, or indicated in a state other than ‘verification pass’, at every issuing entry point in the complete target set (if even one entry point lets either of them through as a ‘verification pass’, the result is nonconforming).
- For the input in which the evidence-level field is missing, the field concerned in the stored and issued result is not filled with a default value or the highest level but remains missing, or the request is rejected.

- The original input is issued normally at every entry point, and the scoring code version of the isolated copy used for the test is the same as the version of the production issuing build.

Record items

- The lists of issuing entry points derived from the two sources, and their union and set difference
- Per-round input differences, identifiers of the target entries, and, for each entry point, whether issuing took place and the rejection reason string
- Actual value of the evidence-level field after the missing-data input was stored
- Scoring code versions of the isolated copy and of the production issuing build

> Caution — **pitfall**: Where the evidence level exists only as a declared string that is not checked against anything and the issuing path trusts that value as it stands — the input with the lowered level is honestly rejected and the test passes, but in actual issuing an entry bearing an L2 label attached to L1 evidence goes out as a ‘verification pass’ without any check.

C.5.2 PAVS-REP-02 — Confirms that no score report is issued under a standard name not registered in the issuing system, that the attribution of the grounds for a verdict is indicated item by item, and that the attribution points to a clause that actually exists.

Test type: injection

Preconditions

- A list of registered standards exists within the issuing system and can be queried at the level of standard name and clause identifier.
- The issuing entry points can be derived from two sources — the interface specification and the path records of the issuing ledger — and can be called on an isolated copy.
- The population of recently issued score reports can be enumerated together with the attributes of issue date, issuing format and scenario type.

Procedure

1. The tester derives the issuing entry points from the two sources and fixes their union as the complete target set.

2. The tester creates issuing requests in which the name of the standard cited as verdict grounds is forged in three variants — a number string that does not exist, the name of an existing standard that is well-formed but not on the registered list, and the number of a withdrawn earlier edition of a registered standard.

3. The tester feeds each of the three variant requests into every entry point in the complete target set, observes for each entry point and each variant whether the request is rejected and the reason, and confirms normal issuing with 1 request of identical structure carrying the name of a registered current standard (placebo control).

4. The tester stratifies the recently issued score reports by issuing format and scenario type, and draws 10 (all, if the total is fewer than 10) with a recorded random seed so that at least 2 are included from each stratum.

5. The tester enumerates all items of the drawn score reports, counts whether each item individually carries a label ‘derived from a reference standard / defined by the issuer itself’, and separately counts the number of score reports that present attribution only as an aggregate ratio.

6. For every item labeled ‘derived from a reference standard’, the tester checks whether that standard name is on the registered list and whether the indicated clause identifier actually exists within that standard.

Pass criteria

- All three forged variant requests are rejected at every issuing entry point in the complete target set, and the request carrying a registered current standard is issued normally (if even one variant gets through at even one entry point, the result is nonconforming).
- In every score report in the sample, every item has an item-level attribution label, and there are 0 score reports that present attribution only as an aggregate ratio.
- For every item labeled ‘derived from a reference standard’, the standard name is on the registered list and the clause identifier actually exists (if there is even 1 check failure, the result is nonconforming).

Record items

- The lists of entry points derived from the two sources, and, for each entry point and each variant, whether the request was rejected and the reason
- The three variant standard strings used for the injection
- Stratification criteria, population scope, random seed and sample size per stratum
- Number of items with a missing label, number of clause-identifier check failures, and identifiers of the score reports concerned

> Caution — **pitfall**: Where standard names are filtered only by rules of form (numbering scheme and prefix) — made-up strings are rejected and the test passes, but unregistered standards of correct form and the numbers of withdrawn earlier editions pass through as they are, and items end up attributed to clauses that do not actually exist.

C.5.3 PAVS-REP-03 — Confirms that a score report discloses not-measured items as a list rather than as a ratio, that it does not count not applicable (N/A) as met, and that a fail cannot be evaded by converting it into not applicable.

Test type: injection

Preconditions

- A score-report generation path that carries the verdict state of each item (pass/fail/not-measured/not applicable) is in operation.
- The run of 1 test scenario can be reproduced in an isolated environment, and measurement channels can be blocked item by item.
- The summary section and the detail section of a score report can each be queried.

Procedure

1. In the test scenario, the tester blocks the measurement channels of 3 items to inject missing data in which no sample exists at all, and does not insert a value of 0.

2. In the same run, the tester declares 2 items ‘not applicable’ and also composes an input that additionally requests that 1 item whose measured result is a fail be converted to ‘not applicable’.

3. The tester generates the score report and observes whether the 3 not-measured items appear in the detail section as a list accompanied by the item identifiers and the reasons for the missing data, and whether the not-measured count is printed together beside the met figure in the summary section.

4. The tester confirms that the items not blocked were measured normally, thereby establishing that the 3 not-measured items are not a by-product of a failure of the whole run.

5. The tester confirms whether the not-applicable items are included in the met count (numerator), whether they are distinguished from the scored items in the denominator, and whether the conversion of a failed item to not applicable is accepted without grounds.

6. The tester regenerates the score report with the original scenario, without blocking or conversion, and confirms that the not-measured list is empty (placebo control).

Pass criteria

- All 3 not-measured items appear in the detail section as a list accompanied by the identifiers and the reasons for the missing data, and the not-measured count is printed beside the met figure in the summary section (a ratio indication alone is nonconforming).
- Not-applicable items are not included in the met count and are distinguished from the scored items in the denominator, and the conversion of an item that failed by measurement to not applicable is rejected or is indicated separately together with the grounds for the conversion.
- The items not blocked were measured normally, and in the score report of the original run the not-measured list is empty.

Record items

- Identifiers of the items into which missing data was injected, the method of blocking the channels, and the measurement state of the items not blocked
- Full text of the not-measured list in the generated score report and the string printed in the summary section
- Figures of the met count and the denominator, and the place where not applicable is handled
- Result of processing the request to convert a failed item to not applicable, and the result of the placebo control

> Caution — **pitfall**: Where the injection of missing data causes the whole run to fail — all items become not-measured at once, so the not-measured list is plausibly filled and the test passes, but whether missing data on an individual channel is folded into 0 and whether not applicable enters the numerator are both left unexamined.

C.5.4 PAVS-REP-04 — Confirms that the self-declaration of a test scenario cannot raise the verdict, that a run resting on self-declaration alone has no issuance authority, and that, where a self-declaration has contributed to the overall verdict, that fact is indicated item by item.

Test type: injection

Preconditions

- A path exists through which a scenario can itself declare a pass or a fail.
- The measurement channels of measured items can be blocked item by item, and the same run output can be re-scored with only the self-declaration value changed.
- The overall verdict and the item-level aggregate (met numerator and denominator) can be queried.

Procedure

1. With all measured items made not-measured, the tester performs a run in which the scenario self-declares ‘pass’, and observes the overall verdict and the **issuance authority** (the entitlement to produce a score report or conformance indication) — this examines whether issuance authority is granted even where the overall verdict is a pass.

2. The tester opens the aggregate of the same round and confirms whether the self-declared item is mixed into the met numerator without distinction from the measured items and pushes up the met rate, or whether its self-declared status is indicated item by item.

3. Keeping the measured output fixed as a single output, the tester re-scores with only the self-declaration value changed to each of three values, ‘pass’, ‘fail’ and ‘no declaration’, and compares the overall verdicts of the three rounds — if the overall verdict of the ‘pass’ round

is better than that of the ‘no declaration’ round, the self-declaration has raised the verdict (the overall verdict of the ‘fail’ round being lower is permitted).

4. In a run in which some of the measured items were measured normally, where the self-declaration contributed to the overall verdict, the tester confirms whether that contribution is indicated in the score report as a sentence or a field.

5. In a round in which a self-declared ‘fail’ and a measured ‘pass’ disagree, the tester confirms that the self-declaration remains in the record and is not silently discarded.

6. The tester compiles into a single table, for each round, the self-declaration value, the states of the measured items, the met numerator and the overall verdict.

Pass criteria

- A run in which all measured items are not-measured has no issuance authority (even if the overall verdict is a pass, no score report or conformance indication is produced), and self-declared items are not mixed into the met numerator without distinction from measured items (separate indication or exclusion from the numerator).
- In the three rounds in which the measured output is fixed and only the self-declaration value is changed, the overall verdict of the ‘pass’ round is no better than that of the ‘no declaration’ round — self-declaration cannot raise the verdict.
- In a run in which a self-declaration contributed to the overall verdict (including where it lowered the verdict), the contribution is explicitly indicated item by item, and a self-declaration that disagrees with the measurement remains in the record.

Record items

- Comparison table of the self-declaration value, the states of the measured items, the met numerator and the overall verdict for each round
- Method of blocking the measurement channels and method of fixing the measured output
- Verbatim string of the contribution indication
- Location where the rounds in which the self-declaration and the measurement disagree are recorded

> Caution — **pitfall**: Where the self-declaration has been demoted to ‘1 evidence item’ but that item enters the met numerator without distinction from the measured items — if all other items are not-measured, that one item makes the met rate 100%, and the test passes on the confirmation that only the promotion of the overall verdict has been blocked. Conversely, an implementation that reads ‘self-declaration cannot move the verdict’ as ignoring even a self-declared fail issues as a pass a run that the scenario itself called a failure (for cases, see the Part 2 Casebook).

C.5.5 PAVS-REP-05 — Confirms that the scope of meaning of a pass verdict is disclosed item by item through boundary notes, that boundary notes are not selectively omitted depending on the verdict state, and that they contain item-specific conditions.

Test type: audit

Preconditions

- The population of recently issued score reports can be enumerated together with issue date, issuing format, scenario type and item verdict states.
- A boundary-note entry exists in the item definitions, and the items to which a boundary note has been assigned can be queried.
- The boundary-note sentences printed in score reports can be extracted item by item.

Procedure

1. The tester stratifies the recently issued score reports by issuing format and scenario type, and draws 10 (all, if the total is fewer than 10) with a recorded random seed so that at least 2 are included from each stratum.

2. The tester enumerates all items of the drawn score reports regardless of verdict state, and counts in a table, for each item, whether a boundary note is present and which of the distinctions ‘runtime guarantee / capability boundary / absence of stimulus’ the note explicitly states.

3. The tester constructs a cross-tabulation of verdict state (pass/fail/not-measured) × boundary note (assigned/printed) and observes whether omissions are concentrated in a particular verdict state.

4. The tester compares the number of items to which a boundary note is assigned in the item definitions with the number of boundary notes actually printed in the score reports and, for each case of difference, confirms whether that item was subject to scoring in the score report concerned.

5. The tester counts the proportion in which the same boundary-note sentence is reused for different items within the sample, to determine whether it is a boilerplate sentence that does not contain item-specific conditions.

6. The tester identifies and records the conditions (verdict state, item type and issuing format) under which differences and skews occur.

Pass criteria

- Every item in the sample is accompanied by its assigned boundary note regardless of verdict state, and each note explicitly states one of the three distinctions.
- In the cross-tabulation, print omissions are not skewed toward a particular verdict state, and the difference between the defined count and the printed count is 0, or every case of difference is individually confirmed as ‘not subject to scoring in that score report’.
- The proportion of identical sentences reused for different items does not exceed half of all boundary notes in the sample.

Record items

- Stratification criteria, population scope, random seed and sample size per stratum
- Table of the presence and distinction type of the boundary note for each item, and the cross-tabulation by verdict state
- Result of comparing the defined count with the printed count, and the confirmation details for each case of difference
- Proportion of reuse of identical sentences and the full text of the reused sentences

> Caution — **pitfall**: Where the same boilerplate sentence is attached as the boundary note to every item — an audit that counts only presence and distinction type passes it as ‘assigned to all’, but no item states the scope of meaning of its own conditions, so the scope of the pass is still left blank to the reader.

C.5.6 PAVS-REP-06 — Confirms whether success-rate-type figures are accompanied by the trial count and uncertainty figures, and whether improvement claims present the null-control result alongside, with a statement that the conditions are the same.

Test type: audit

Preconditions

- The population of results documents can be enumerated in full for each external issuing channel.

- Whether there is an automated check that detects the accompanying elements of numerical reporting, and its execution path, have been identified.
- The list of expressions to be regarded as improvement or superiority claims is fixed and recorded before the test.

Procedure

1. The tester enumerates all results documents of the most recent 3 months for each external issuing channel as the population, and draws 8 with a recorded random seed so that at least 1 is included from each channel.

2. The tester enumerates all figures presented as percentages or ratios in the drawn documents, and counts in a table, for each figure, whether it is accompanied by the trial count and an uncertainty figure (the actual numbers of the upper and lower bounds of an interval or of the standard error) — textual indications such as ‘approximate’ or ‘roughly’ are not counted as uncertainty.

3. The tester enumerates all improvement or superiority claim sentences that fall under the fixed list of expressions, and counts whether the null-control result is presented alongside, together with the trial count, in the same table or the same section, and whether it is stated that the two values were obtained under the same conditions.

4. Where an automated check exists, the tester feeds into the check a test document from which the trial count and uncertainty have been removed and observes whether a flag is raised, and confirms with 1 document in which the accompanying elements are complete that no false flag is raised (placebo control).

5. Where no automated check exists or it is not hooked into any automated execution point, the tester records that fact as a deficiency.

6. For the entire population outside the sample, the tester mechanically counts only whether a trial count is indicated beside each figure, and checks whether the sample result and the trend across the entire population diverge.

Pass criteria

- In the sample, there are 0 percentage or ratio figures not accompanied by both the trial count and an uncertainty figure, and in the full-population check the deficiency rate outside the sample is not higher than the deficiency rate of the sample.
- For every improvement or superiority claim in the sample, the null-control result is presented alongside together with the trial count, with an explicit statement that the conditions are the same.
- An automated check exists, flags the deficient document and does not falsely flag the complete document (if no automated check exists or it is not wired, this criterion is recorded not as met but as not-measured).

Record items

- Population scope per issuing channel, random seed and sample size, and the fixed list of improvement and superiority expressions
- Table of whether each figure is accompanied by the trial count and an uncertainty figure, and the result of the full-population check outside the sample
- For each improvement claim, whether the null control is presented alongside and whether the conditions are stated
- Outputs of the 2 rounds of feeding into the automated check, or the record of the absence or non-wiring of the automated check

> Caution — **pitfall**: Where an uncertainty field exists in the template but is always empty or filled with words such as ‘approximate’ — an audit that counts only the presence of the field passes, but the issued documents print the success rate of a single batch beside the trial count, with no interval at all, as if it were a definitive figure.

C.5.7 PAVS-REP-07 — Confirms whether the known simplifications, approximations and dormant defects of the scoring model are entered in the register together with their validity range and direction of distortion, whether approximations outside the register are looked up from both the outputs and the change history, and whether resolution or a qualifying indication precedes any issuing outside the validity range.

Test type: audit

Preconditions

- An official register in which dormant defects, simplifications and approximations are entered exists, its entries are managed by state transitions rather than by deletion, and all of them can be enumerated.
- The change history of the scoring engine and the scenario conditions of recently issued score reports can be queried.
- The measurement provenance of each item (measured/commanded value/default value/estimate) can be queried exhaustively in score-report outputs.

Procedure

1. The tester enumerates all entries classified in the register as ‘dormant defect, simplification or approximation’, and counts in a table, for each entry, whether the validity range and the direction of distortion (optimistic/pessimistic) are stated.

2. The tester takes the recent change history of the scoring engine as the population and draws 10 (all, if the total is fewer than 10) with a recorded random seed, placing the changes not accompanied by a register entry in a separate stratum and drawing at least 5 from that stratum.

3. For each drawn change, the tester counts the number of additions or modifications of constants, thresholds, clamps, default values and substitute paths, and counts how many of them are not entered in the register.

4. To catch approximations outside the register from the output side, the tester enumerates all items in recently issued score reports whose measurement provenance is indicated as ‘commanded value, default value or estimate’, and checks whether each of them has a corresponding entry in the register.

5. For each entry with a stated validity range, the tester queries whether score reports involving conditions outside that range were issued after the entry date and, if any were issued, confirms whether a qualifying indication is carried or whether the resolution record precedes the issue date.

6. The tester confirms that the previous statements of entries in the register that have transitioned to ‘resolved’ are preserved.

Pass criteria

- Every entry in the register states both its validity range and its direction of distortion, and the previous statements of entries that have transitioned to ‘resolved’ are preserved.
- In the change-history sample there are 0 unentered approximations, and every item that appears in the outputs with ‘commanded value, default value or estimate’ provenance has a corresponding entry in the register.
- A qualifying indication or a preceding resolution record exists for every issue involving conditions outside the validity range.

Record items

- Complete list of entries in the register and the number of missing statements, and the preservation status of the previous statements of entries transitioned to resolved
- Change-history population, random seed and sample size per stratum, and the number of unentered approximations
- Result of the measurement-provenance check and list of items without a corresponding entry
- Result of the query on out-of-range issues, identifiers of the score reports concerned, and verbatim string of the qualifying indication

> Caution — **pitfall**: Where ‘complete enumeration’ is counted only within the register — simplifications never put in the register cannot in principle be looked up, so the test passes on the grounds that the few entries in the register are complete, and unless a cross-check is made against the two sources outside the register — the measurement provenance of the outputs and the change history — unentered approximations go on producing silent optimistic passes.

C.5.8 PAVS-REP-08 — Confirms whether a claim of ‘completion’ is accompanied by a formal self-report of what is achieved, deferred and omitted, whether that self-report is checked against the list of deliverables at initiation, unmodified after the fact, and whether, where anything is not met, the result is indicated not as completion but as partial achievement.

Test type: audit

Preconditions

- The population of results documents claiming completion or closure can be enumerated by period and by author.
- The list of deliverables at initiation corresponding to each document is preserved separately, and its recording time and subsequent modification history can be queried.
- The vocabulary list to be regarded as completion-type and deferral-type expressions is fixed and recorded before the test.

Procedure

1. The tester enumerates, as the population, the results documents of the most recent 6 months that contain the fixed completion-type vocabulary, stratifies them by author, and draws 10 (all, if the total is fewer than 10) with a recorded random seed.

2. For each document, the tester confirms whether the recording time of the corresponding list of deliverables at initiation precedes the issue time of the completion document and, if there is a history of the list having been modified after initiation, records that difference entry by entry in a table.

3. For each document, the tester counts whether a self-report section of what is achieved, deferred and omitted exists, and whether that section is in the form of a list of entries rather than a summary narrative.

4. The tester checks the self-report section of each document entry by entry against the original list of deliverables at initiation, and counts the number of deliverables missing from the self-report.

5. The tester confirms whether the achieved count and the denominator in the self-report section are arithmetically consistent with the completion claim in the body text, and counts whether documents with 1 or more deferred or omitted entries are indicated with completion-type vocabulary.

6. The tester compiles the comparison results for each document into a single table and identifies the cases in which the indication and the actual fulfillment status disagree.

Pass criteria

- Every document in the sample has a self-report section of what is achieved, deferred and omitted in the form of a list of entries, and the recording time of the list of deliverables at initiation precedes the issue time of the completion document.
- There are 0 omissions from the self-report against the original list of deliverables at initiation, and where the list was modified after initiation, that difference is stated in the document.
- Among documents with 1 or more deferred or omitted entries, there are 0 indicated with completion-type vocabulary, and the counts in the self-report section are arithmetically consistent with the claim in the body text.

Record items

- Stratification criteria, population period, random seed and sample size, and the fixed list of completion-type and deferral-type vocabulary
- Recording time and modification history of the list of deliverables at initiation, and the difference table
- For each document, the presence and form of the self-report section, and the number of omissions found by the check against the deliverables
- List of disagreements between the completion indication and the actual fulfillment status

> Caution — **pitfall**: Where the list of deliverables at initiation can be modified even after the fact — if the list is quietly amended so that only the achieved entries remain, omissions from the self-report become structurally 0, and although it is the shrinking of the denominator that justifies the declaration of completion, the test passes as it stands on the grounds that ‘a separate basis for comparison exists’.

C.5.9 PAVS-REP-09 — Confirms whether tampering, deletion, reordering and tail truncation are actually detected in the ledger of issued score reports, and whether resistance to tampering in which the integrity values have also been rewritten is backed by an anchor outside the ledger.

Test type: injection

Preconditions

- A test copy of the ledger isolated from the issued records and a ledger verifier are available, and the test copy contains at least 5 issue records.
- Whether the same verifier can also be run in the persistent layer in which the actual issued records are stored is identified.
- Whether the length of the ledger or its latest integrity value is recorded and fixed in an independent layer outside the ledger is identified.

Procedure

1. The tester runs the verifier once on the test copy and confirms the intact state as the baseline.

2. The tester runs the verifier on a copy in which only the body (the verdict result field) of 1 intermediate record is changed and the integrity value is left unchanged.

3. The tester makes a copy in which 1 intermediate record is deleted, a copy in which the order of two records is swapped and a copy in which the tail record is cut off, and runs the verifier on each.

4. The tester makes a fifth copy in which the body is changed and the integrity values are recomputed by the prescribed method, runs the verifier on it, and determines whether detection is achieved only through the internal linkage of the ledger or through comparison with an anchor outside the ledger.

5. The tester records the verifier verdict and the flagged location for each of the five variants, and repeats the same procedure on a copy in the persistent layer in which the actual issued records are stored.

6. The tester runs the verifier again on an untampered copy and confirms that there are no false detections (placebo control).

Pass criteria

- In all four variants — body tampering, deletion, reordering and tail truncation — the verifier produces a fail and flags the location, and for the untampered copy it reports intact.
- The fifth variant, with recomputed integrity values, is detected or, where it is not detected, the absence of an independent anchor outside the ledger is recorded as a deficiency and the test result for the tamper-resistance requirement of this clause is recorded as ‘inconclusive’ (C.0.3).
- The results of the five variants are reproduced identically in the persistent layer in which the actual issued records are stored as well.

Record items

- Specific manipulations of the five injection variants and the identifiers of the target records
- Verifier output for each round (verdict and flagged location) and identification of the ledger layer on which verification was performed
- Whether an independent anchor outside the ledger exists, and its update interval
- Baseline and placebo-control results, and the verifier version

> Caution — **pitfall**: If the verifier checks only the internal linkage of the integrity values and has no anchor outside the ledger — when a party that can also rewrite the integrity values changes the body or cuts off the tail, the inside of the ledger is fully consistent, so all four injection variants are detected and the test passes, while actual tampering is reported as intact without exception.

C.5.10 PAVS-REP-10 — Confirms whether the issue of a score report without a declaration of the fact of involvement in adaptation or training and of the scope of independence is rejected, whether a non-declaration is not filled with a default value, and whether the ‘neutral’ indication is restricted to cases with no involvement in adaptation.

Test type: injection

Preconditions

- The score report template defines fields for the party involved in adaptation and for the scope of independence, and the version in which those fields were introduced is identified.
- The issuing entry points can be derived from two sources — the interface specification and the path records of the issuing ledger — and can be called on an isolated copy.
- The actual values of the two fields stored in the issued records can be inspected after issue.

Procedure

1. The tester derives the issuing entry points from the two sources and fixes their union as the complete target set.

2. The tester feeds an issue request with the fields for the party involved in adaptation and the scope of independence left empty into every entry point in the complete target set, and observes for each entry point whether the request is rejected.

3. For each entry point that did not reject the request, the tester opens the stored issued record, confirms what values remain in the two fields, and determines whether ‘no adaptation’, ‘neutral’ or the like has been filled in automatically.

4. The tester feeds in a request that fills in the party involved in adaptation as ‘the issuer itself’ while indicating ‘neutral’, and a request that fills in the same organizational unit as the issuer, or an affiliate, as the adaptation party while indicating ‘neutral’, and observes whether each is rejected or the neutral indication is removed.

5. The tester feeds in a request filled in as no involvement in adaptation (third-party adaptation or no adaptation) and confirms that it is issued normally (placebo control).

6. Across all records issued since the field-introduction version, the tester queries the distribution of values of the two fields and counts the number of non-declarations and the number of values that appear to have been filled in automatically.

Pass criteria

- The request with both fields empty is rejected at every issuing entry point in the complete target set and, where there is a path that does not reject it, the stored fields are not filled with default values but remain missing.
- The requests indicating ‘neutral’ that name the issuer itself, the same organizational unit as the issuer or an affiliate as the adaptation party are all rejected or have the neutral indication removed.
- There are 0 non-declarations among the records issued since the field-introduction version, and the request with no involvement in adaptation is issued normally.

Record items

- List of entry points derived from the two sources, and for each entry point whether the request was rejected and the reason
- Field-value composition of the 4 kinds of test request and the actual field values of the stored issued records
- Field-introduction version, and the field-value distribution and number of non-declarations of the records issued since then
- Point at which the rule restricting the neutral indication is applied, and the placebo-control result

> Caution — **pitfall**: If the issuing path has a default value that reads a non-declaration as ‘no adaptation’ — an empty request is not rejected but is issued filled in as ‘neutral’, and since the fields are filled in when queried after issue, the tester misreads a non-declaration count of 0 as ‘a declaration exists’ and passes the test.

C.5.11 PAVS-REP-11 — Confirms that verdicts restored after the fact (backfilled) carry a provenance indication that distinguishes them from real-time verdicts, are not created on evidence that is not checked against the original run outputs, and do not infringe existing verdicts or the subsequent aggregates and results documents.

Test type: injection

Preconditions

- 1 record whose verdict has already been settled and 1 record whose verdict is empty are prepared in a test copy.
- The backfill procedure can be run on an isolated copy, and the backfill evidence material can be replaced with material unrelated to the original run outputs.
- The provenance field can be queried at each stage — the verdict list, the aggregate and the results document.

Procedure

1. The tester runs a backfill on the record whose verdict is empty, on the basis of the original run outputs, and observes whether the produced verdict carries a provenance value indicating that it is a backfill together with the restoration time and an evidence reference (placebo control — this determines that a normal backfill is not rejected).

2. The tester attempts a backfill on the same record with evidence that is unrelated to, or not checked against, the original run outputs, and observes whether it is rejected or remains not-measured.

3. The tester runs a backfill on the record whose verdict has already been settled, and observes whether the existing verdict value and the settlement time change.

4. The tester changes the evidence material so that a result different from the settled record is produced, reruns the backfill, and confirms whether the overwrite is blocked and whether the attempt itself is left in the history.

5. The tester queries the backfilled verdicts and the real-time verdicts at each of the three stages — the verdict list, the aggregate and the results document — and confirms whether the two are counted and indicated separately on the basis of the provenance field alone.

6. The tester compiles the verdict value, provenance and time for each round and the query results for each stage into one table.

Pass criteria

- The backfilled verdict carries a provenance value distinct from real-time verdicts and a reference to the restoration evidence, and a backfill on evidence not checked against the original run outputs is rejected or remains not-measured.
- The settled existing verdict remains unchanged in both value and time after the backfill is run, and the overwrite attempt is left in the history.
- At all three stages — the verdict list, the aggregate and the results document — backfilled verdicts and real-time verdicts are counted and indicated separately.

Record items

- Verdict value, provenance and time before and after the backfill is run
- Identification of the 3 kinds of evidence material (original-run evidence, unrelated evidence and conflicting evidence) and the handling result of each attempt
- Result of the overwrite attempt, the blocking point and the location where the history is recorded
- Results of the separate-counting queries at each stage — the verdict list, the aggregate and the results document

> Caution — **pitfall**: If the provenance indication is attached only in the backfill executor and the original records and is merged at the aggregate and results-document stages — a test that opens only the original records confirms the distinction and passes, but in issued documents and statistics backfilled verdicts are counted in the same cell as measured verdicts and inflate the proportion of measured verdicts.

C.5.12 PAVS-REP-12 — Confirms that the overall status of the score report does not report as ‘normal’ a result in which a safety anomaly metric declared by the range operator is other than 0 or missing.

Test type: injection

Preconditions

- The list of safety anomaly metrics exists as a per-profile declaration (data) and can be queried in full, and it can be confirmed whether that declaration lies outside the scoring engine code.

- Result inputs can be composed for each profile in an isolated environment to generate score reports, and metric values can also be made missing.
- The set of values that the overall status field can take is enumerated in advance.

Procedure

1. The tester establishes the profile population as the union of ‘all profiles on the declaration list’ and ‘all profiles that actually appear in recent issues’, and enumerates the list of anomaly metrics for each profile.

2. For each profile, the tester generates a score report from a result input in which 1 anomaly metric is injected with a value other than 0, and observes whether the overall status field drops to a value other than ‘normal’ in the enumerated value set.

3. The tester injects the same metric separately with the value of the smallest representable unit below the threshold (where a threshold is applied) and as missing, and observes whether the overall status is other than ‘normal’ in those two cases as well.

4. The tester identifies the metrics that are on the declaration list but are not registered as scoring items, and confirms whether the overall status is pushed down when those metrics are injected as well.

5. The tester performs the same injection with a profile of a type not on the declaration list, and observes whether it is silently handled under the rules of another profile instead or is indicated as not-measured or rejected.

6. The tester generates a score report from a result in which all anomaly metrics are 0 and nothing is missing, and confirms whether the overall status is reported as ‘normal’ (placebo control).

Pass criteria

- Across the entire profile union, in every case in which an anomaly metric is injected in the three forms — a value other than 0, the minimum value below the threshold and missing — the overall status is not reported as ‘normal’.
- The overall status is not ‘normal’ when a metric not registered as an item is injected either, and an input for a profile not on the declaration list is not handled under another profile's rules but is indicated as not-measured or rejected.
- For a result in which all anomaly metrics are 0 and nothing is missing, the overall status is reported as ‘normal’, and it is confirmed that the anomaly-metric declaration exists as data outside the scoring engine code.

Record items

- Basis for deriving the profile union, and the anomaly-metric list and injected values for each profile
- Table of overall-status results by profile and by injection form
- List of metrics not registered as items, and the location of the anomaly-metric declaration
- Handling result of the unknown-profile input, and the placebo-control result

> Caution — **pitfall**: If there is a path on which an anomaly metric is folded into 0 when it is missing — injection of a value other than 0 honestly yields ‘not normal’, so the test passes for all profiles, but in actual operation with a measurement channel disconnected the anomaly metric becomes missing and the overall status continues to be reported as ‘normal’.

C.5.13 PAVS-REP-13 — Confirms whether results documents are generated directly from the verdict of record (whether there is no parallel layer that recomputes verdicts), and whether not-measured is not folded away in the document but actually appears in the measurement reliability section.

Test type: injection

Preconditions

- The verdict of record and the results-document generation path are identified separately and can be run on an isolated copy.
- The export surfaces of results documents can be derived from two sources — the template definitions and the surface records of the recent issuing ledger.
- The item-level indications and the measurement reliability section can each be extracted from a results document.

Procedure

1. The tester derives the export surfaces from the two sources and fixes their union as the complete target set.

2. In the verdict of record, the tester changes 1 item ‘pass→fail’, another 1 item ‘fail→pass’ and yet another 1 item ‘pass→not-measured’, and regenerates the results document with all other inputs left unchanged.

3. In the regenerated document, the tester observes whether the indications of the three items have changed in the same direction as the verdict of record or have remained at their previous values or at values recomputed against its own thresholds — in particular, the tester determines whether the item changed to ‘not-measured’ is folded into a pass in the document.

4. The tester makes the verdict of record unavailable for query and then generates the document, and observes whether the document fills the gap with its own threshold table or ends in generation failure or a not-measured indication.

5. The tester repeats the two preceding injections for every export surface in the complete target set and compiles the results for each surface into one table.

6. The tester opens the measurement reliability section of the generated document and checks whether the identifier of the injected not-measured item actually appears in that section's not-measured list.

Pass criteria

- The result of changing the verdicts of record in three directions is reflected in the same direction on every export surface in the complete target set, and the item changed to ‘not-measured’ is not indicated as a pass on any surface.
- When the verdict of record cannot be queried, no surface produces a verdict by substitute computation (generation failure or a not-measured indication).
- The measurement reliability section exists, and the identifiers of the injected not-measured items all appear in that section's not-measured list.

Record items

- Lists of export surfaces derived from the two sources, and their union and difference
- The injected diffs to the verdict of record (three directions) and the identifiers of the target items
- Table of document indication results for each surface, and the behavior of each surface in the absence of the verdict of record
- The actual list in the measurement reliability section and the result of checking it against the not-measured identifiers

> Caution — **pitfall**: If the list of export surfaces is taken only from the template definitions — a secondary surface that is not in the definitions and leaves traces only in the ledger is not exposed even if it recomputes verdicts with its own threshold table, so a state in which the same test result has a different verdict in each document passes as ‘directly linked to the verdict of record’.

C.5.14 PAVS-REP-14 — Confirms whether, when a measurement contradicting a boundary note occurs, that note is automatically flagged, whether the flagging mechanism is wired into an automatic execution point that is actually triggered, and whether it also applies to newly assigned boundary notes.

Test type: injection

Preconditions

- At least 2 items to which boundary notes are assigned exist, spanning different item types.
- The flagging mechanism and the automatic execution point to which it is attached are identified, and the recent execution history of that point can be queried.
- 1 new boundary note can be assigned to an item definition immediately before the test.

Procedure

1. The tester queries the recent execution history of the automatic execution point to confirm whether there is a round in which it was triggered without human intervention and, if there is none, records that fact as a deficiency.

2. The tester takes as targets 1 item whose boundary note reads ‘absence of stimulus’, 1 item of another type and 1 item to which a boundary note was newly assigned immediately before the test, and records the original sentences verbatim.

3. The tester creates test results into which measured results that actually decide pass or fail for each of the three items are injected.

4. With no one running the flagging mechanism directly, the tester triggers the declared automatic execution point once, and observes whether the three items are flagged in its output as ‘stale boundary notes’.

5. The tester injects measurements consistent with the boundary notes (no contradiction), triggers the same automatic execution point, and confirms that no flagging occurs (placebo control).

6. When flagging occurs, the tester records the termination state of the automatic execution point, attempts a subsequent issue in that state, and observes whether the issue is blocked or proceeds with the boundary note of the item concerned indicated as ‘unconfirmed’.

Pass criteria

- All three items into which contradicting measurements were injected are flagged in the output of the automatic execution point (the tester does not run the flagging mechanism separately, and the new boundary note assigned immediately before the test is flagged as well).
- No flagging occurs for measurements without contradiction, and a history of the automatic execution point being triggered without human intervention exists in the recent execution records.
- The subsequent issue after flagging is blocked or, if the score report is issued, the boundary note of the item concerned is indicated in the score report in the ‘unconfirmed’ state.

Record items

- Identifiers of the 3 target items and the original boundary-note sentences (including the 1 newly assigned)
- Content of the injected contradicting measurements, and the identification and full output of the automatic execution point
- Recent execution history of the automatic execution point and the rounds triggered without human intervention

- Placebo-control result, termination state, and handling result of the subsequent issue attempt

> Caution — **pitfall**: If the flagging mechanism is fixed to a specific list of item identifiers — a test using only existing items flags all of them and passes, but boundary notes assigned later are not added to that list, so stale notes remain unflagged forever and continue to be printed in score reports.

C.5.15 PAVS-REP-15 — Confirms that the score report indicates the physical-hardware verification stage of the simulation model and the status of its correlation evidence, and that, where there is no evidence, that fact remains as ‘none’ and the indication itself is not omitted.

Test type: injection

Preconditions

- The results-document generation path can be run on an isolated copy.
- The place at which the verification stage of the model (self-consistency / cross-comparison / comparison with measured data) is declared is identified.
- The presence or absence of correlation evidence and its source can be derived from the verdict of record or an equivalent record.

Procedure

1. The tester generates a score report with a model that **has** correlation evidence, and confirms whether the stage indication and the evidence source actually appear in the document.

2. The tester removes the correlation-evidence record of the same model, regenerates the score report, and observes whether the indication remains as ‘none’ or **the indication itself disappears** — the latter is the failure this test targets.

3. The tester lowers the verification-stage declaration from ‘comparison with measured data’ to ‘self-consistency’, regenerates, and confirms whether the indication in the document follows downward in the same direction.

4. Placebo control: the tester changes a field unrelated to the correlation evidence (for example, the test date), regenerates, and confirms that the stage indication does not shift.

5. The tester repeats step 2 for every export surface (for each document template, if there are two or more).

Pass criteria

- The three-stage distinction appears in the document, and the ‘none’ indication remains when there is no correlation evidence.
- There are 0 surfaces on which the indication is omitted.
- The stage indication does not change under the placebo manipulation.

Record items

- Presence or absence of the indication on each surface
- Document differences before and after the removal of the evidence
- Result of the placebo manipulation
- Source path of the stage declaration

> Caution — **pitfall**: If the stage cell is hard-coded into the document template as a **constant string**, the cell remains even when the evidence is removed and passes step 2 — step 3 (regeneration with the stage lowered) discriminates this. If the stage indication does not move in step 3 either, that cell is not a valid indication, and the result of this test is ‘inconclusive’.

C.6 OPS — Operations test methods (10)

C.6.1 PAVS-OPS-01 — Confirms whether failed verification and measurement approaches are registered in the registry, whether a registry lookup is actually performed before the start of a new investigation or retry and returns the relevant entries, and whether a start that omits the lookup is blocked on every start path.

Test type: audit

Preconditions

- The dead-end registry is maintained at a designated location, and the auditor is granted read access to all of its entries.
- The start register, which records the start of new investigations and retries, and a third source independent of it (execution-resource occupancy records or output generation records) can each be queried.
- The period under audit, the population freeze time and the random seed are agreed and recorded before the audit begins.

Procedure

1. The tester freezes, at the same time, all entries of the registry, the starts in the start register and the traces of execution launch in the third source, counts the ‘unregistered starts’ that are present in the third source but have no corresponding entry in the start register, and adds them to the population.

2. From the population after this addition (the whole period under audit or the most recent 50 cases, whichever contains more cases), the tester draws 12 cases at random with the recorded random seed and, if fewer than 3 of the cases drawn are classified as retries, makes a supplementary draw from a stratum consisting of retry cases only until there are 3.

3. For each case drawn, the tester confirms the existence and the chronological order of a registry lookup record made before the start time (lookup time, the person who performed the lookup, and the identifier of the entry looked up or the basis for the judgment ‘no matching entry’).

4. Separately from the sample, the auditor designates as positive controls 3 starts whose subject is the same target and the same approach as an ‘valid’ entry in the registry, and confirms whether the lookup record of each such start actually returned the identifier of the corresponding registry entry.

5. If there are two or more start paths, the tester finalizes the list of paths; on each path, witnessed by the auditor, the tester makes 1 attempt at a start that omits the lookup step and observes whether it is blocked or returned and the blocking point, and, as a placebo control, attempts on the same path 1 start in which the lookup is performed normally and confirms that it is not blocked.

6. The tester draws in the reverse direction 5 entries in ‘valid’ status from the registry, and checks whether the start register or the third source contains any case in which the approach concerned was performed again without a lookup after its registration.

Pass criteria

- For every case in the sample, a registry lookup record preceding the start time exists and states the identifier of the entry looked up or the basis for the judgment ‘no matching entry’, and there are 0 unregistered starts against the third source.
- In the 3 positive controls, the lookup record returned the identifier of the corresponding registry entry (0 cases closed as ‘no matching entry’ despite a matching subject), and there are 0 re-performances without lookup in the 5 reverse-direction comparisons.

- On every finalized start path, the start that omits the lookup ends in blocking or return, and the normal start of the placebo control is not blocked.

Record items

- Population freeze time, the case counts of the three sources and the number of unregistered starts
- Random seed and the list of identifiers of the sampled cases (including indication of the stratum supplement)
- For each sample, whether a lookup record exists and the determination of the order of the lookup time and the start time, and the entry identifiers returned by the 3 positive controls
- List of start paths, whether the witnessed demonstration on each path was blocked and the blocking point, and the result of the placebo control

> Caution — **pitfall**: Even if the lookup is a formal procedure whose record leaves only ‘looked up’ and which returns ‘no matching entry’ to any query, counting only whether the record is entered makes every case met; unless starts that have a registry entry with a matching subject are designated as positive controls and it is confirmed whether that entry is actually returned, an operation whose lookup is nominal only passes.

C.6.2 PAVS-OPS-02 — Confirms whether registry entries operate the four-state status lifecycle through actual transitions, and whether, on the correction or withdrawal of a verdict, the original record is not deleted or concealed, even on the path that users see, but is preserved linked to the correction record.

Test type: audit

Preconditions

- The dead-end registry, the verdict ledger and the status change history (including times) are provided to the auditor with read access.
- Copies of results documents already issued can be queried independently of the verdict ledger, and the auditor can use the query path of the results-document user role.
- The period under audit and the random seed are agreed and recorded before the audit begins.

Procedure

1. The tester aggregates the status values of all registry entries and computes together the distribution of counts of ‘valid’, ‘resolved’, ‘partially resolved’ and ‘undetermined’, the numbers of status values outside the defined set and of blanks, and the number of status transitions that actually occurred within the period under audit.

2. From the population of entries whose status is ‘resolved’ or ‘partially resolved’, the tester draws 8 (all, if the population has fewer than 8) with the recorded random seed, and confirms whether a resolution basis (the identifier of a refutation test or of a prerequisite fix) is linked to the entry and whether the time of the basis record precedes the time of the status change.

3. For all corrections and withdrawals of verdicts in the period under audit, the tester confirms whether the original verdict record remains in a queryable state and whether the correction record refers to the identifier of the original record.

4. The tester opens 1 correction not through the auditor-only query but through the query path of the results-document user role, and observes whether the original content and the link to the subsequent correction are returned together.

5. The tester draws at random 5 copies of results documents already issued, compares their verdict values with the values in the current verdict ledger, and confirms whether a correction record is linked to each mismatch.

6. If there are 0 status transitions within the period under audit, the tester records the lifecycle-operation part as ‘not-measured’ and leaves the reason.

Pass criteria

- Status values exist only within the four defined states, blanks and values outside the defined set number 0, and at least 1 status transition is observed within the period under audit (if there are 0, the result is recorded not as a pass but as not-measured).
- Every sampled resolved or partially resolved entry has a resolution-basis identifier linked to it and the time of the basis precedes the time of the status change, and for all corrections and withdrawals the user-role query returns the original content together with the correction link.
- In the comparison of the issued copies with the verdict ledger, there are 0 mismatches to which no correction record is linked.

Record items

- Number of entries per status, number of blanks and of values outside the defined set, and number of status transitions within the period under audit
- Random seed, and for the 8 samples the identifiers, the resolution-basis identifiers and the determination of chronological order
- Complete list of corrections and withdrawals and the results of the user-role query (whether the original content is returned)
- Results of comparing the 5 issued copies with the verdict ledger, and the number of mismatches without a linked correction

> Caution — **pitfall**: Even if the original record remains only in the auditor-only query and the results documents seen by users show only the post-correction values, an audit that opens only the ledger counts this as ‘preserved’; unless the user-role query and the comparison of issued copies are performed, silent retroactive modification is counted as preservation.

C.6.3 PAVS-OPS-03 — Confirms whether, for each registered failed approach, retry preconditions are stated on a standing basis in a form whose truth or falsity can be determined by observation, and whether a precondition that has already been met is not left unattended so that the retry remains sealed off.

Test type: audit

Preconditions

- The list of all registry entries and the statement change history of each entry (including times) are frozen and provided to the auditor.
- At least 1 entry whose status is ‘valid’ or ‘partially resolved’ exists, and starts classified as retries can be identified in the start register.
- The audit notice time and the random seed are recorded before the audit begins.

Procedure

1. The tester freezes the entries in ‘valid’ or ‘partially resolved’ status as a subpopulation, draws 10 (all, if the population has fewer than 10) with the recorded random seed, and transcribes their precondition statements verbatim.

2. The tester classifies each precondition as (a) a fact to be proven first, (b) a prerequisite fix or (c) neither, and determines observability by the following rule — a statement is classified as observable only when it ① names the object of observation (measurement channel,

output or record identifier) and ② specifies the direction of satisfaction (threshold comparison, presence/absence or true/false), so that the auditor can determine whether it is satisfied without further inquiry to the operator; if either of the two requirements is absent, it is classified as not observable.

3. The tester compares the statement time of the precondition of each sample with the audit notice time and, for entries that have had a retry, with the retry start time, counts separately the statements newly created or revised after the audit notice, and excludes them from the determination of whether the requirement is met.

4. For all retry starts actually performed in the period under audit, the tester checks whether evidence showing that the preconditions of the entry concerned were met is linked to the start record.

5. The auditor determines, by observing in the stated direction of satisfaction, the entries in the sample whose preconditions have already been met, and confirms whether each such entry has a retry start record or a statement of the planned resolution time.

6. If any entry has its precondition stated as ‘none’, the tester confirms whether the reason (for example, that the approach concerned is not viable in principle) is stated alongside.

Pass criteria

- Every sampled entry states a precondition or a ‘none’ with its reason stated alongside, and there are 0 statements newly created or revised after the audit notice.
- There are 0 entries classified as not observable because the object of observation or the direction of satisfaction is not specified.
- Every entry with an already met precondition has a retry start record or a statement of the planned resolution time, and evidence of satisfaction is linked to all retry starts.

Record items

- Number of cases in the subpopulation, random seed and sample identifiers
- For each sample, the verbatim precondition, the (a)/(b)/(c) classification and the result of the observability determination
- Number of cases classified as not observable, and the element of the determination rule applied (the requirement lacking, ① or ②)
- Number of entries already met and their follow-up (retry start or planned time), and number of revisions after the audit notice

> Caution — **pitfall**: If the precondition fields are revised in bulk into observable sentences after the audit notice is received, an inspection of the text alone finds every case met; unless the statement times are compared with the audit notice time and the retry start times, statements that existed only during the audit period are counted as standing operation.

C.6.4 PAVS-OPS-04 — Confirms whether test outputs and decisions are recorded on a standing basis in the designated single source of truth, whether prior reading before a start is performed on the version in effect at that time, and whether stale records are corrected within the set deadline.

Test type: audit

Preconditions

- The official record location for each domain (single source of truth) and the correction deadline for stale records are designated in a document.
- The auditor has read access to all of the record system, the test execution register and the client communication records.

- The creation time and the correction history of records are preserved with times, and the audit notice time is recorded.

Procedure

1. The tester freezes a population from each of two sources independent of the record system — the completed tests in the test execution register and the decisions in the client communication records — and draws at random, with the recorded random seed, 12 completed tests and 6 decisions.

2. The tester confirms whether the decisions and outputs of each sample can be retrieved from the designated record location, and counts separately the cases that exist only in non-designated locations (personal records, one-off conversation records and the like).

3. The tester compares the record creation time of each sample with the completion or decision time and with the audit notice time, and counts separately the records created after the audit notice.

4. In the start record of each sample, the tester confirms the existence and chronological order of traces of prior reading before the start time (the identifier and version of the document read beforehand), and checks whether the recorded version agrees with the version that was in effect at the start time.

5. The tester selects the 5 most recent entries in the correction history of the designated record location, computes the interval between the time at which the stale record was found and the correction time, and compares it with the designated correction deadline.

6. The tester selects 3 of the samples, compares the description in the record system with 3 corresponding values in the verdict of record and, if there is a mismatch, confirms whether a correction history is linked to that mismatch.

Pass criteria

- The decisions and outputs of every sample drawn from the two sources can be retrieved from the designated record location, and the cases that existed only in non-designated locations and the records created after the audit notice each number 0.
- Every sample has a prior-reading record preceding the start time, and the recorded version of the document read beforehand agrees with the version in effect at the start time.
- The discovery–correction interval of the 5 correction-history entries is within the designated deadline, and there are 0 mismatches between the records and the verdict of record or a correction history is linked to every mismatch.

Record items

- Names, freeze times and case counts of the two population sources, and the random seed and sample identifiers
- Number of cases that existed only in non-designated locations, and their identifiers
- Number of records created after the audit notice and number of prior-reading version mismatches
- Discovery time, correction time and interval of the 5 correction-history entries, and number of cases exceeding the deadline

> Caution — **pitfall**: If, after the audit notice, the execution register is swept and missing records are registered retroactively, the retrieval rate becomes 100% for all cases by definition; unless the record creation times are compared with the completion times and the audit notice time, an operation that does not record on a standing basis passes.

C.6.5 PAVS-OPS-05 — Confirms whether the procedural sequence and confirmation points specified by the client were observed, whether the mandatory major steps set by the specification proceeded after a confirmation that refers to each step individually, and whether the confirmation did not change verdict values.

Test type: audit

Preconditions

- The engagement document specifies the procedural sequence, the confirmation points and the pre-agreed scale threshold.
- Confirmation requests and confirmation responses are preserved with their times, and the auditor can query a source showing that those times have not been altered after the fact (an append-only record or a client-side copy).
- The time and reason of changes to verdict values can be queried in the verdict history.

Procedure

1. The tester freezes the engagements in the period under audit as the population and draws 8 (all, if the population has fewer than 8) with the recorded random seed; tests that exceeded the scale threshold are identified independently, not from the engagement records but from the test execution register, and their engagements are placed in a separate stratum and covered in full.

2. The tester extracts the list of major steps from each sampled engagement; the mandatory major steps set by the specification (proceeding with the verdict-settlement procedure, starting a test that exceeds the scale threshold, declaring completion, and adding a test outside the scope of the engagement) are included in every case, irrespective of whether the engagement document enumerates them.

3. For each major step, the tester collects the confirmation request time, the confirmation response time and the step execution time, compares their order, and cross-checks the response time against the client-side copy or the append-only record.

4. The tester classifies whether each confirmation response refers to the step concerned individually or reuses a blanket approval covering a period or scope, and counts the cases in which the same response identifier is cited as the basis for two or more major steps.

5. The tester classifies the content of the confirmation responses into ‘confirmation of procedural progress’ and ‘instruction to change verdict content’ and, if any response is classified as the latter, traces in the verdict history of that case whether the verdict value was changed.

6. The tester counts and records the number of mandatory major steps for which no confirmation point is designated in the engagement document.

Pass criteria

- For every major step of the samples, including the mandatory major steps, the chronological order ‘confirmation request → confirmation response → step execution’ holds, and the response time is cross-checked against an independent source.
- The confirmation response for each major step refers to that step individually, and there are 0 cases in which one response identifier is reused as the basis for two or more major steps.
- There are 0 cases in which a verdict value was changed on the basis of a confirmation response.

Record items

- Identifiers of the sampled engagements and the list of major steps for each engagement (including indication of the mandatory steps)

- For each step, the confirmation request, response and execution times, the determination of their order, and the source used to cross-check the response time
- Number of starts exceeding the scale threshold identified from the execution register, and whether each had a prior confirmation
- Number of blanket-approval reuses, number of responses classified as ‘instruction to change verdict content’, and the tracing results

> Caution — **pitfall**: If tests exceeding the scale threshold are identified only from the engagement records, threshold-exceeding cases executed without prior confirmation never enter the stratum in the first place; unless the threshold-exceeding cases are reconstructed independently from the execution register, large-scale runs that skipped the confirmation points are excluded from the sample in principle.

C.6.6 PAVS-OPS-06 — Confirms by defect injection that the automatic gates of test intake and execution actually fire on the basis of measured values as declared and distinguish missing from the value 0, and that the reasons for rejections and holds reach a channel that the client can query.

Test type: injection

Preconditions

- A test intake and execution path configured identically to the operational path can be used for audit purposes, and there is no dedicated marker or bypass path that distinguishes injected requests from operational requests.
- The list of gate conditions and the measurement channel and measurement provenance referred to by each condition are declared in a document.
- The receipt records of rejection and hold notifications can be queried with a client-role account.

Procedure

1. For every declared gate condition, the tester classifies the measurement provenance of the referenced value as measured, commanded value, default value or estimate, and compiles together the conditions based on values that are not measured and whether each such condition can by itself produce a ‘met’ determination.

2. For all gate conditions (if there are 11 or more conditions, 10 drawn with the recorded random seed, plus all core conditions), the tester injects, one at a time, a request that violates only that one condition with the remaining conditions filled with normal values, and then traces by request identifier the state transitions and the notification entries.

3. For 3 of the conditions declared as referring to measurement, the tester injects requests in which the two values are made to diverge so that the commanded value is normal and the measured value is a violation, and observes whether the gate rejects or holds following the measured value or lets the request through following the commanded value.

4. As a placebo control, the tester injects a normal request that meets all gate conditions, and observes whether the test reaches execution without rejection or hold.

5. For the same measurement channel, the tester injects a request with a ‘measured value of 0’ and a request with ‘missing’, and observes whether the former is handled as a determination according to the value and the latter as a rejection, hold or not-measured, so that the two results are distinguished from each other.

6. The tester queries the reasons for the rejection or hold of each injected request not with auditor privileges but with the client-role account, and confirms whether the identifier of the injected condition and the reason reached the notification entries.

Pass criteria

- Every violation-injection request ends in rejection or hold, and the identifier of the injected condition and the reason are confirmed in the client-role query; the normal request of the placebo control is not rejected and reaches execution.
- In the 3 provenance-mismatch injections the gate fires following the measured value, and there are 0 cases in which a condition based on a commanded value, default value or estimate by itself produced a ‘met’ determination.
- The handling results of the measured-value-0 request and the missing request are distinguished from each other, and the missing request is not folded into ‘met’.

Record items

- List of gate conditions and the result of classifying the measurement provenance of each condition, and the method of selecting the injection targets (whether all were covered, and the random seed)
- Identifiers of the injected requests and the injected content (violated condition, the values made to diverge and the missing channel)
- Observed state transitions for each request and the verbatim notification entries confirmed by the client-role query
- Identifier of the placebo-control request and whether it reached execution

> Caution — **pitfall**: Even if a gate is documented as referring to a measurement channel, if it actually reads the commanded value carried in the request, it rejects normally in ordinary violation injections as well and is counted as ‘fires on a measured basis’; unless an injection that makes the commanded value and the measured value diverge determines which value was followed, a gate without a measured basis passes.

C.6.7 PAVS-OPS-07 — Confirms whether unverified ‘impossible’ claims that are restricting the scope of functions or tests are registered in the list without omission, and whether they are being resolved within the deadline by being refuted or confirmed through reproduction tests under the same conditions.

Test type: audit

Preconditions

- A list of the reasons restricting the scope of functions or tests is maintained, and each entry indicates its basis classification (measured basis/unverified claim) and its planned resolution time.
- The auditor can query the identifiers and execution records of the tests cited as the basis for a restriction, as well as the reasons for test rejections and holds and the reasons for ‘not-measured’ verdicts.
- The period under audit and the random seed are recorded before the audit begins.

Procedure

1. The tester freezes the population of currently effective restriction entries, and forms a subpopulation by combining the entries whose basis classification is ‘unverified claim’ with the entries whose basis classification is blank.

2. From sources outside the restriction list — reasons for test rejections and holds, reasons for ‘not-measured’ verdicts, and statements of exclusion from application in results documents — the tester collects the restriction reasons that appear repeatedly, compares them with the restriction list, counts the unregistered restriction reasons and adds them to the subpopulation.

3. For every entry marked ‘measured basis’, the tester confirms whether the cited test identifier can actually be retrieved and whether the result of that test supports the restriction

claim, and adds to the subpopulation any entry that cannot be retrieved or whose result does not support it.

4. From the subpopulation, the tester draws 6 (all, if the subpopulation has fewer than 6) with the recorded random seed, checks the history of resolution attempts (identifier, execution time and result of the refutation or confirmation test), and, for entries without such a history, confirms whether the retry precondition and the planned resolution time are stated.

5. The tester counts the entries whose planned resolution time has already passed, and computes the number of them with no statement of a resolution attempt or of a reason for resetting the planned time after it passed.

6. The tester selects 3 entries marked ‘resolved’ and checks, divided into four elements, whether the basis test was performed with the same stimulus, the same measurement channel, the same configuration and the same verdict threshold as the restriction claim.

Pass criteria

- There are 0 unregistered restriction reasons found in sources outside the restriction list.
- Every sample is classified as either ‘resolved by refutation or confirmation’ or ‘unresolved, but an observable resolution precondition and a planned time that has not yet passed are stated’, and there are 0 entries left without update after the planned time has passed.
- The basis tests of the 3 entries marked as resolved agree on all four comparison elements, and their results support the direction of resolution.

Record items

- Number of restriction entries in the population and their distribution by basis classification, and the list of unregistered restriction reasons collected from sources outside the list
- Random seed, sample identifiers and the resolution-history identifiers for each sample
- Number of cases by reason for addition to the subpopulation (cited test not retrievable, result mismatch, basis blank)
- Number of entries whose planned resolution time has passed, and the number of those left without update

> Caution — **pitfall**: If a restriction is not registered in the list but is indicated repeatedly only as the rejection reason of individual tests or as the reason for ‘not-measured’ verdicts, that restriction never enters the population; unless restriction reasons are re-collected from sources outside the list, unverified ‘impossible’ claims that are actually narrowing the scope remain invisible to the audit.

C.6.8 PAVS-OPS-08 — Confirms whether launched tests reached normal termination and actual reclamation of resources, and whether the indiscriminate launching of tests before a hypothesis is settled is restrained irrespective of the way identifiers are assigned.

Test type: audit

Preconditions

- The test execution register records the launch time, termination time and termination reason, and the identifier of the hypothesis or of an equivalent investigation unit.
- The occupancy status of execution resources can be queried per test identifier, and that query is performed through a path independent of the execution register.
- The resource-reclamation grace period and the upper limit on consecutive launches for the same hypothesis are agreed and recorded before the audit begins.

Procedure

1. The tester freezes all launches in the period under audit as the population, and tallies their termination states as ‘normal termination’, ‘aborted’ and ‘status unknown’.

2. The tester takes as the sample all cases classified as aborted or status unknown, plus 10 drawn at random from the normally terminated cases with the recorded random seed.

3. For each sample, the tester performs a resource-side query after the point obtained by adding the agreed reclamation grace period to the termination time, and confirms whether any execution resources occupied under that test identifier remain.

4. At 2 different points in time during the audit period, the tester compares the resource occupancy list with the ‘in progress’ list of the execution register, and counts separately the occupancies with no corresponding entry in the register (phantom occupancies) and the reverse-direction mismatches in which the register shows a test in progress but there is no occupancy.

5. The tester computes the reuse distribution of hypothesis identifiers (number of tests per identifier), and counts the launches with identical SUT, scenario and stimulus configuration that were assigned different hypothesis identifiers.

6. On both bases — by hypothesis identifier and by ‘identical configuration’ grouping — the tester computes the maximum chain length of tests launched consecutively before the result of the preceding test was settled.

Pass criteria

- There are 0 cases of residual occupancy after the reclamation grace period across all samples, and 0 phantom occupancies in each of the comparisons at 2 different points in time.
- There are 0 launches classified as status unknown or a reclamation action record is linked to every such case, and a termination handling record is linked to every case in which the register shows a test in progress but there is no occupancy.
- The larger of the maximum chain lengths computed on the two bases is at or below the agreed upper limit, or a parallel-design reason (control groups, repeated samples, etc.) is stated for every case exceeding the upper limit.

Record items

- Number of cases by termination state, population freeze time, and the agreed reclamation grace period and upper limit on consecutive launches
- Random seed, sample identifiers and the result of the residual-occupancy query for each sample
- Lists of phantom occupancies at the 2 points in time (resource identifier and occupancy start time) and the number of reverse-direction mismatches
- Reuse distribution of hypothesis identifiers and the maximum chain lengths computed on the two bases

> Caution — **pitfall**: If a new hypothesis identifier is assigned to every launch, the chain length is always 1 and no indiscriminate launching of tests is counted; unless chains are also computed on a basis that groups launches with identical stimulus configuration, this test can be evaded by the way identifiers are assigned alone.

C.6.9 PAVS-OPS-09 — Confirms whether the adoption rationale of the core components of the verification infrastructure is recorded, whether an SUT of a new family reaches the issuance of a verdict solely through the addition of a declaration document, without a configuration change, and whether this does not change the verdicts of existing SUTs.

Test type: audit

Preconditions

- The list of core components and the record of the adoption rationale of each component are maintained, and the configuration change history can be queried together with the time and scope of each change.
- At least 1 new SUT was accepted in the period under audit (if there is no acceptance history, the acceptance-path part of this test is recorded as not-measured).
- The SUT to be used in the witnessed demonstration and its declaration document can be designated and provided by the auditor or the SUT supplier.

Procedure

1. The tester draws 5 from the list of core components with the recorded random seed, and confirms whether each states the identification of an externally verified algorithm or tool and the reason for its adoption, or the reason for choosing an in-house implementation together with grounds of unsuitability that name the tool in question.

2. For all new SUTs accepted in the period under audit, the tester compares the identifiers of the added declaration documents with the configuration change history of the same period, and counts the changes, within a pre-agreed window before and after the acceptance time, that contain branches or exception handling referring to the SUT concerned.

3. The target of the witnessed demonstration is designated by the auditor but is chosen from a family for which there are 0 changes in the configuration history, across the whole period under audit, that refer to the name or identifier of that SUT or its family; the declaration document used is one prepared not by the operator but by the auditor or the SUT supplier.

4. The tester performs the demonstration under witness, observes whether, solely through the addition of the declaration document, the test is executed, measured values are produced on the measurement channels and a verdict (pass, fail or not-measured) is issued, and confirms from the history the configuration changes that occurred in the process.

5. Immediately before and immediately after the demonstration, the tester runs 1 existing SUT once in the same scenario on each occasion, and confirms whether the verdict and the measured values designated for comparison have not changed.

6. If a configuration change occurred during the demonstration, the tester classifies it by scope of change as either a change referring to the SUT concerned or a common defect fix, and records the basis of the judgment.

Pass criteria

- All 5 samples state an adoption rationale, and those that chose an in-house implementation include grounds of unsuitability naming the tool in question.
- In the witnessed demonstration, a verdict is issued solely through the addition of the declaration document (if the verdict is not-measured, its reason is indicated together with the identifier of the missing measurement channel), there are 0 configuration changes referring to the SUT concerned, and in the regression comparison the verdict and comparison measured values of the existing SUT do not change.
- Across all acceptances in the period under audit, there are 0 changes containing SUT-specific branches or exception handling, or the reason and a resolution plan are stated for those that occurred.

Record items

- Random seed and a summary of the adoption rationale of the 5 sampled components
- For each acceptance, the declaration document identifier, the configuration change window compared and the number of changes classified as dedicated branches
- Basis for the family judgment of the witnessed demonstration target, the author of the declaration document, and the verdict and measured values issued
- Verdicts and comparison measured values of the 2 regression comparisons

> Caution — **pitfall**: If the operator prepares the demonstration target and its declaration document, a variant of an existing family into which dedicated code has already been incorporated is selected and 0 configuration changes are observed; unless the absence of changes for the target family is verified from the configuration history and the author of the declaration document is placed outside the operator, the declarative acceptance capability is observed as broader than it actually is.

C.6.10 PAVS-OPS-10 — Confirms whether issued score reports were produced with the version actually used of a scoring system conforming to this specification, and whether its conformance claim does not stop at self-reporting but is supported by evidence of automated-check execution targeting the same version.

Test type: audit

Preconditions

- The auditor can query the ledger of issued score reports, and each score report records the identifier and version of the scoring system used.
- The operational records designate, for each scoring system, the storage location and the stored version of its Implementation Conformance Statement (ICS), and the times of version changes of the scoring system can be queried.
- The version of the scoring system actually used by the run corresponding to each score report can be queried in the test execution register.

Procedure

1. The tester freezes the population of score reports issued in the period under audit, stratifies it by the identifier and version of the scoring system used, and draws at least 2 per stratum and at least 10 in total (2 per stratum if 2 times the number of strata exceeds 10) with the recorded random seed. If the population has fewer than 10, all are taken.

2. The tester checks whether the ICS corresponding to the scoring system identifier and version of each sample can be retrieved from the operational records, and whether the version of that ICS was in effect at the time the score report was issued.

3. For each sample, the tester compares the version stated in the score report with the version actually used as recorded in the execution register, and counts the mismatches.

4. The tester identifies the assertions corresponding to the items used by each sampled score report and, for every automated-check assertion among them that the ICS marks as ‘met’, confirms whether automated-check execution evidence (execution time, target version and result summary) is attached and whether its target version is the same as the version of the ICS.

5. The tester reads the ICS's list of not-met assertions and its indications of whether the core assertions are met, and confirms whether the sampled score reports used items affected by not-met assertions and, if so, whether that fact and the scope of impact (identifiers of the affected items and verdicts) are indicated in the score report.

6. The tester finds the times of version changes of the scoring system, and counts the reports issued after a change that cite the ICS of the previous version as the basis for conformance.

Pass criteria

- For every sample, the version stated in the score report, the version actually used according to the execution register and the version of the retrieved ICS agree with one another, and an ICS in effect at the time of issue can be retrieved from the operational records.
- Automated-check execution evidence targeting the same version is attached to every automated-check assertion that corresponds to the items used by the samples and that the ICS marks as met.
- There are 0 score reports issued with a scoring system whose core assertions are not met, or every such score report indicates that fact and the identifiers of the affected items and verdicts; and there are 0 reports issued after a version change that cite the ICS of the previous version.

Record items

- Stratification criteria, number of cases per stratum, random seed and identifiers of the sampled score reports
- For each sample, the result of comparing the three versions (stated in the score report, actually used according to the execution register, and that of the ICS)
- Number of automated-check assertions marked as met, and the numbers with execution evidence attached and with a matching target version
- The ICS's list of not-met assertions and core-assertion indications, the numbers of reports issued before and after the version change time, and the number citing the previous version

> Caution — **pitfall**: Writing down even one not-met assertion makes the list no longer ‘blank’, so that exemption from the whole requirement for automated-check execution evidence can be obtained; unless the evidence requirement is tied not to whether the list is blank but to ‘every automated-check assertion marked as met’, a self-reported ICS passes as the basis for conformance merely by reporting one harmless assertion as not met.

C.7 CNF — Conformance test methods (5)

C.7.1 PAVS-CNF-01 — Confirms that the conformance claim text has the four elements of the prescribed form, discloses every assertion not met without omission, and does not profess conformance while a ★core assertion is not met or is reported as met without a test record.

Test type: audit

Preconditions

- The auditor can access the list of public channels and the issue register of the auditee organization, and has been granted the authority to collect conformance-claim surfaces directly from those channels.
- The auditor has obtained the complete list of Annex A assertions, the list of ★core assertions and the applicable-class table of the claimed version directly from the authoritative copy of the specification's publishing body (copies provided by the auditee organization are not used as the reference for comparison).
- The Implementation Conformance Statement (ICS) of the same version and the same class, and the test records for each assertion (test identifier, date of performance and determination), can be viewed.

Procedure

1. The auditor defines the population as the union of “conformance claims issued or posted in the 12 months immediately preceding the audit reference date” and “conformance claims being posted or distributed as of the audit reference date, regardless of when they were issued”, and includes not only issued documents but also public postings, proposal materials and promotional materials, down to short professions carried within titles, badges, diagrams and images. The auditor fixes the population as the union of the list collected directly by the auditor and the list submitted by the organization, and counts separately, as “unreported surfaces”, the surfaces that were not on the list submitted by the organization.

2. The auditor composes the sample by stratified random draw with a pre-published random seed, 5 from each surface type (issued documents, public postings, proposal materials and promotional materials) (all of that type, if the population of the type is fewer than 5). In a type in which unreported surfaces exist, the auditor draws at least half of that type's sample from the unreported surfaces.

3. On the face of each sampled case, the auditor checks whether the four elements — version, class, list of assertions met and list of assertions not met — are stated. An indication reading “Assertions not met: none” is regarded as the element being present, and only a case in which the element itself is absent from the face is counted as an omission. The auditor also confirms, against the list of issued editions of the specification's publishing body, whether the version indicated is an issued edition that actually exists and had not been withdrawn as of the audit reference date, and tallies the cases for which this does not hold as “invalid version”.

4. The auditor compares the list of assertions not met of each sampled case with the ICS of the same version and the same class, and tallies as “concealed not met” those assertions determined as not met in the ICS (including indications that report the assertion as only partly met) that do not appear in the list of assertions not met in the claim text. For a sample for which comparison is impossible because no ICS is attached, the determination is recorded as not-measured and is not counted as a pass.

5. For each sampled case, the auditor checks the ICS determinations of all ★core assertions and, for each assertion reported as “met”, checks whether a test record of that assertion exists and whether the recorded determination agrees with “met”. A “met” for which there is no test record or whose recorded determination differs is tallied as “unsupported met” and is treated as equivalent to not met in the subsequent determinations.

6. The auditor reads the entire face of each sampled case (including the wording in the body, titles, badges and images) and confirms, quoting the actual wording, whether vocabulary professing conformance — such as “conformant”, “compliant” and “met” — is used. A case in which only a distinguishing expression (for example, “aligned”) is used on its own is not counted as a violation, but a case in which a distinguishing expression and vocabulary professing conformance appear side by side on the same surface is counted as a violation. A profession of conformance in a state of element omission, invalid version, concealed not met or core not met (including unsupported met) is recorded as a violation without asking whether it is an editorial omission or a deliberate omission, and a corrected version presented during the audit is recorded separately together with the date of correction but does not replace the determination on the original claim text.

Pass criteria

- On the face of every sampled case, all four elements — version, class, list of assertions met and list of assertions not met — are present, and the version indicated is a valid issued edition (0 element omissions and 0 invalid versions).
- In every sampled case there are 0 concealed-not-met assertions, and there are 0 samples treated as not-measured because comparison with the ICS was impossible.

- For any version and class in which even one ★core assertion is not met or is an unsupported met, vocabulary professing conformance is not observed in any sampled case (including where it appears side by side with a distinguishing expression).

Record items

- Method of fixing the population, population size, stratum size by surface type, number of unreported surfaces, draw seed and draw date
- For each sample, whether each of the four elements is present and the names of omitted elements, and the version indicated and the result of confirming its validity
- List of identifiers of concealed-not-met assertions computed as the set difference of the claim text against the ICS, and the list of samples treated as not-measured
- For each ★core assertion, the ICS determination, whether a test record exists and the recorded determination; the identifiers of unsupported-met assertions; and, for each sample, the actual wording of the professing vocabulary quoted

> Caution — **pitfall**: If the “claim list” submitted by the organization is taken as the population and only the face of the ICS is compared, the audit accepts as true a self-reported “met” that has no test record, and the single word “conformant” carried on a public posting outside the form falls outside the sample — as a result, an organization whose restraint applies only to its issued documents and an organization that has never once performed a test receive the same pass.

C.7.2 PAVS-CNF-02 — Confirms that a checklist of all assertions of Annex A is attached to the conformance claim identifier by identifier, that not applicable is limited to a mismatch of the applicable class, and that “met” determinations are supported by test records.

Test type: audit

Preconditions

- The conformance claim texts of the same seed and the same sample as **PAVS-CNF-01** and the ICSs attached to them are secured as pairs.
- The auditor has obtained the complete list of Annex A assertions, the applicable-class table and the list of withdrawn numbers of the claimed version directly from the authoritative copy of the specification's publishing body.
- The test records (test identifier, date of performance and determination) on which each row of the ICS is based can be viewed at the time of the audit.

Procedure

1. Using the same seed and the same sample as **PAVS-CNF-01**, the auditor confirms whether an ICS is attached to each claim text and can actually be viewed at the time of the audit. A case that has only an indication of attachment or a reference but cannot be viewed is counted as not attached. The auditor also confirms whether the target version, the class and the reference date of preparation are stated on the face of the ICS, and tallies as “reference-date defect” the cases whose reference date is later than the issue date of the claim text or precedes it by 12 months or more.

2. The auditor compares the set of assertion identifiers of the attached ICS with the set of assertion identifiers of Annex A of that version by two-way set difference, and tallies separately the omitted assertion identifiers and the assertion identifiers not in Annex A (including withdrawn numbers of deleted assertions). The comparison is made identifier by identifier, not by counting rows alone.

3. The auditor confirms whether the determination value of each row is one of the three values met, not met and not applicable, and separately tallies, together with their wording, indications outside the three values, such as blanks, “under review” and “partially met”.

4. For every row indicated as not applicable, the auditor checks back, using the class table obtained by the auditor, whether the Annex A applicable class of that assertion actually differs from the claimed class, and confirms whether the wording of a reason is stated alongside. A row indicated as not applicable although the applicable class is the same as the claimed class is tallied as “unjustified not applicable”.

5. From the “met” rows of each ICS, the auditor draws 10 at random with a pre-published random seed (all, if there are fewer than 10 met rows), with every met row corresponding to a ★core assertion forcibly included in the sampling. For each drawn row, the auditor checks whether a test record of that assertion exists and whether the recorded determination agrees with “met”, and tallies an absence of the record or a disagreement of determination as “unsupported met”.

6. For each row tallied as unjustified not applicable or unsupported met, the auditor confirms whether the determination of **PAVS-CNF-01** changes when that row is reverted to not met (whether that assertion is ★core, and whether the list of assertions not met grows), and records this as well.

Pass criteria

- For every sampled case, an ICS is attached and can be viewed, its target version, class and reference date are stated, there are 0 reference-date defects, and the set of assertion identifiers of the ICS agrees exactly with the set of assertion identifiers of Annex A of that version (0 omitted and 0 in excess).
- The determination value of every row is one of the three values met, not met and not applicable, there are 0 blanks, and for every not-applicable row the mismatch of the applicable class is confirmed by the class table and a reason is stated alongside (0 unjustified not applicable).
- For every sampled met row, a test record exists and the recorded determination agrees with “met” (0 unsupported met).

Record items

- For each sample, whether the ICS is attached and can be viewed, the target version, class and reference date of preparation stated on its face, and whether there is a reference-date defect
- Lists of omitted assertion identifiers and excess assertion identifiers computed by two-way set difference, and the number and wording of rows with indications outside the three values
- List of not-applicable rows and, for each row, the applicable class, the claimed class and the wording of the reason stated alongside, and the list of unjustified not-applicable rows
- Seed and sample count of the met-row sampling, and for each row the test record identifier, date of performance and recorded determination, and the list of identifiers of unsupported-met assertions

> Caution — **pitfall**: An ICS in which every assertion is filled in as “met” without omission looks the most complete in an audit of form, but unless those met entries are traced back to test records, an organization that has never once tested receives the highest score — and if, on top of this, a substitution that moves not-met assertions to “not applicable” to shrink the denominator is layered on, an ICS with neither omissions nor blanks passes while its measured basis is 0.

C.7.3 PAVS-CNF-03 — Confirms that, when a profile attempts to relax the requirements of Clauses 5–10 of the specification, the production registration and application path actually fires a rejection, that relaxation is not achieved through a bypass under a different name or through a change after registration either, and that there is no relaxation in definitions already in application.

Test type: injection

Preconditions

- The operational path for defining and registering profiles and the path for applying registered objects to scoring can be accessed with a test account, and it has been confirmed that the two paths are identical to the production path that produces actual results documents.
- The list of requirements of Clauses 5–10 to be maintained and the determination criteria for each relaxation type are declared before the test, and a substantive determination rule — that any registered object that selects a set of items and is applied to scoring is regarded as a profile regardless of its name — is declared along with them.
- The definitions for the test and their scoring results are given a test marker and are not reflected in issued results documents.

Procedure

1. The tester registers, through the production registration path, 1 normal profile that only selects items and inherits the requirements of Clauses 5–10 unchanged, executes scoring once with that profile, and confirms, as the baseline, that acceptance of the registration, application to scoring and production of verdicts are observed in sequence. If this baseline is not observed, the tester records the determination of this test as not-measured and does not perform the subsequent steps, and in no case is it counted as a pass.

2. The tester attempts to register 5 profiles, each through the same production registration path, into each of which only one of the 5 types of requirement relaxation is injected at a time — ① a verdict rule that folds not-measured into a pass; ② a provenance relaxation that has an item requiring measurement scored with values of non-measured provenance (commanded value, default value or estimate); ③ loosely overwriting the pass threshold of an assertion on the profile side; ④ suppression of mandatory reporting elements (indication of measurement provenance and counting of not-measured); ⑤ exemption from the negative-control and null-control requirements.

3. The tester additionally injects 2 bypass types — ⑥ attempting registration and application by putting the same content as relaxation ① into a scoring-configuration registered object under a name other than profile; ⑦ passing registration with a normal definition, then changing the definition of the registered object to relaxation ① and applying it to scoring without re-verification.

4. For each of the 7 injections, the tester confirms whether the point at which the rejection fired is the production registration and application path or the pre-validation of a test-auxiliary path, and records whether the wording of the rejection reason carries the clause identifier of the violated requirement. A case caught only in pre-validation but accepted by the production path is counted as not rejected.

5. For each injected case not rejected at the registration or application stage, the tester executes actual scoring once and, for the same test result, places the verdicts produced with that definition and the verdicts produced with the base requirements side by side, item identifier by item identifier, and tallies the shifts in the more lenient direction (not-measured→pass, fail→pass and an increase in the number of passing items).

6. The tester registers 2 placebo controls — 1 selection-only definition that only reduces the number of items and leaves the requirements untouched, and 1 definition that reduces the items while making the pass threshold stricter. The tester then builds the population as the union of the registration history and the registered objects applied in the test execution records, applies the same relaxation determination rule retrospectively to all of them, and tallies the definitions that constitute relaxation.

Pass criteria

- Registration, application and verdict production of the baseline are observed, all 7 injections of the 5 relaxation types and 2 bypass types are rejected on the production registration or application path, and the rejection reason states the clause identifier of the violated requirement.
- Neither of the 2 placebo controls (selection-only and threshold-tightened) is rejected, and both are registered and applied (0 misfires).
- There are 0 lenient verdict shifts in the injected cases not rejected, and in the exhaustive retrospective tally over the union of the registration history and the execution records there are 0 definitions that constitute relaxation.

Record items

- Original text of the definitions of the 7 injections and, for each, the registration and application result, the wording of the rejection reason and the point at which the rejection fired (production path or pre-validation)
- Evidence of registration, application and verdict production for the baseline definition, and the test marker assigned
- Verdict comparison table, item identifier by item identifier, for the injected cases not rejected, and the number of lenient shifts
- Size of the retrospective-tally population by source (registration history and execution records) and the number of relaxations detected, and the registration and application results of the 2 placebo controls

> Caution — **pitfall**: A system that has no profile mechanism or has 0 registered objects has neither anything to inject into nor a population to examine retrospectively, and so passes with the appearance of “0 relaxations”, but this is not a pass; it is not-measured — conversely, even where the mechanism exists, if the rejection fires only in the pre-validation of a test-auxiliary path, a relaxation that comes in through production is applied to scoring as it is, and only “rejected” remains in the test record.

C.7.4 PAVS-CNF-04 — Confirms that a recipient who encounters a claim amounting to partial conformance can reach the list of assertions not met from the same surface without logging in or making a separate request, and that no conformance indication hiding the existence of assertions not met exists on any surface or in any language.

Test type: audit

Preconditions

- The auditor can collect conformance-claim surfaces directly from the public channels of the auditee organization, and has been granted the authority to fix the population as the union of that collected list and the list submitted by the organization.
- The recipient access condition of each surface (public viewing, viewing after registration, provision on request or non-public) and its posting language are identified.
- The reachability test can be performed in a new viewing session that is not logged in.

Procedure

1. The auditor fixes the population by the same rule as **PAVS-CNF-01** (the union of what was issued in the immediately preceding 12 months and what is being posted as of the audit reference date, including what the auditor collected directly), except that a surface whose same content is posted in two or more languages is counted as a separate case for each language. The auditor makes a stratified random draw with a pre-published random seed, 5 from each surface type (all of that type, if the population of the type is fewer than 5).

2. The auditor counts the number of assertions not met in the ICS corresponding to each sampled case, and divides the samples into partial-conformance samples, with 1 or more assertions not met, and full-conformance samples, with 0 assertions not met. For a sample whose corresponding ICS cannot be identified, the determination is recorded as not-measured and is not counted as a pass.

3. The auditor reads the face of each sampled case as it stands (including the wording in the body, titles, badges and images), and regards the existence of assertions not met as appearing on the face if one of the following three applies — ① an enumeration of the identifiers of assertions not met; ② an indication reading “Assertions not met: none” (limited to full-conformance samples — a “none” indication on a partial-conformance sample is tallied separately as a false indication); ③ a statement that assertions not met exist, together with a pointer to the location of their list. An indication that falls under none of the three (for example, “conformant” or “compliant” standing alone without a qualifier) is tallied as a concealing indication (a “none” indication on a partial-conformance sample is tallied only as a false indication).

4. For each partial-conformance sample, the auditor moves to the list of assertions not met in a new viewing session that is not logged in, taking only that surface as the starting point, and records the number of steps taken and the time of observation. Here one step means one move following a reference carried on that surface, and a case in which reaching the list takes more than one step, or requires account registration, a separate request, a non-public annex or a reply after contact, is counted as not reached.

5. For samples indicated as full conformance as well, the auditor compares them against the ICS to confirm whether there are actually 0 assertions not met, and separately tallies the cases indicated as full conformance although assertions not met exist. A copy of each sample surface as of the time of observation is preserved, and a surface modified during the audit is determined on the pre-modification copy, with the date of modification recorded separately.

Pass criteria

- For every partial-conformance sample, the list of assertions not met is reached within one step by viewing without login (0 not reached and 0 not-measured due to inability to identify the ICS).
- The face of every partial-conformance sample falls under one of the two forms (enumeration of identifiers, or a statement of existence with a location pointer) (0 concealing indications and 0 false “none” indications).
- For every sample indicated as full conformance, there are 0 assertions not met in the ICS (0 false indications of full conformance).

Record items

- Population size and sample size by surface type and by language, draw seed and draw date, and the number of surfaces not reported by the organization
- For each sample, the number of assertions not met according to the ICS, the original wording of the indication carried on the surface, the time of observation and the identifier of the preserved copy
- For each sample, the path to the list of assertions not met, the number of steps, whether the viewing session was logged in, and whether the list was reached

- List of surfaces on which concealing indications, false “none” indications or false indications of full conformance were observed, and their wording

> Caution — **pitfall**: If it is confirmed only that the list of assertions not met “exists” and not that it “is reached”, a claim that has moved the list into a document provided on request passes as it is — and if, in addition, the auditor confirms reaching in a session already logged in, a list that is visible only after logging in is also recorded as “reached”, and an organization that has written the assertions not met into only one language version of a multilingual surface, unless the sample draw distinguishes languages, sends the single word “conformant” left in the remaining languages outside the audit.

C.7.5 PAVS-CNF-05 — Confirms that every profile actually applied to scoring is declared in a public document with its name, item list, selection rationale and excluded items, that claims of profile conformance state the identifier and version, and that changes of definition are revealed through the version.

Test type: audit

Preconditions

- The identifiers and versions of the applied profiles can be exhaustively extracted from the test execution records for a specified period, and the number of runs in which no profile identifier is recorded can be counted as well.
- The posting location of the public profile declaration documents can be viewed without login, and the auditor has obtained, from the authoritative copy of the specification's publishing body, the list identifying the standard item set as of each version.

Procedure

1. The auditor extracts the identifiers and versions of the applied profiles from all test execution records of the 12 months immediately preceding the audit reference date and builds the “applied set”. In doing so, the auditor tallies runs in which the profile identifier or version is not recorded as “unattributable runs” and, if their number is 1 or more, records the scope of the unattributable runs as not-measured. The auditor builds the “declared set” from the public declaration documents and compares the two sets in both directions, in units that bind the identifier and the version together. The population is built from the execution records, not from the list of declaration documents.

2. For each declared profile, the auditor confirms on the face of the document the presence of the four elements — name, item list, selection rationale and excluded items — and confirms whether the declaration document is accessible by viewing without login. A declaration available only on request or only for viewing after registration is counted as not public.

3. The auditor compares, item identifier by item identifier, whether the declared list of excluded items agrees with the set difference “standard item set as of that version − item list of that profile”, and tallies the identifiers that disagree. The auditor then confirms whether the selection rationale is stated with reference to the identifiers of the excluded items or to item group symbols, and tallies as “rationale unlinked” the cases that contain only a general description without reference to the excluded items.

4. The auditor separately tallies the “undeclared applications”, which are only in the applied set, and the “unapplied declarations”, which are only in the declared set. A case whose identifier is declared but whose version applied in a run is not in the declared set is counted as an undeclared application.

5. For all claim texts professing profile conformance (if there are more than 20, 20 drawn with the same seed as **PAVS-CNF-01**), the auditor confirms whether the profile identifier

and version are stated on the face, and whether the declaration document corresponding to that identifier and version can be viewed without login at the time of the audit.

6. For profiles whose definition was changed during the sample period, the auditor confirms whether the version was raised upon the change, whether the declaration document of the pre-change version remains viewable, and whether the records of runs scored with the pre-change version do not point to the post-change version.

Pass criteria

- Every case in the applied set is included in the declared set in units that bind the identifier and the version together, and there are 0 unattributable runs (0 undeclared applications and 0 not-measured).
- Every declared profile can be viewed without login and has all four elements, the declared list of excluded items agrees with the set difference item identifier by item identifier, and there are 0 rationale-unlinked cases.
- Every sampled claim of profile conformance states the identifier and version and the declaration document of that version can be viewed, and in every case of a definition change the version is incremented and the declaration document of the pre-change version remains viewable.

Record items

- Original lists of the applied set and the declared set (identifiers and versions), the result of the two-way comparison and the number of unattributable runs
- For each declaration, whether the four elements are present and the viewing condition, and the item identifiers that disagreed in the excluded-item comparison and the list of rationale-unlinked cases
- Wording of the identifier and version indications in the sampled claims of profile conformance, and the results of viewing the declaration documents
- For each definition change, the date of change, whether the version was incremented, whether the pre-change version can be viewed, and the version to which the pre-change execution records point

> Caution — **pitfall**: If the population is drawn from the list of declaration documents, the declaration of every case holds trivially; but even if it is drawn from the execution records, the result is the same when those records leave no profile identifier and version, so that “many unattributable runs” is turned into a pass of “0 undeclared applications” — moreover, in an audit that compares only identifiers without counting versions, a profile whose definition alone was quietly changed while its declaration document was left as it was remains conformant.

Annex D (informative) — Comparison table of predecessor concepts

Several mechanisms of this specification are **inherited from concepts established by existing standards**. This annex states the inherited concepts and the matters added by this specification, distinguishing between them. If the lineage is not stated, inheritance is mistaken for reinvention; if it is stated, the scope of the additions becomes clear.

Mechanism of this specification	Predecessor concept (source)	What this specification adds
3-valued verdict — pass · fail · not-measured (Clause 7)	pass / fail / **inconclusive** — conformance testing methodology of the ISO/IEC 9646 series	Not-measured **is folded into neither the denominator nor the numerator of either the pass rate or the fail rate.** If all items are missing, the pass rate is produced not as 0% or

		100% but as **‘not computable’**. Exception — an absence of measurement of a mandatory item may be counted as a fail for the reason ‘absence of measurement’, and even in that case the count of not-measured results is retained separately (**PAVS-VDT-01**)
Implementation Conformance Statement (ICS) — 11.3	ICS / PICS proforma — ISO/IEC 9646 series	None (both form and name inherited). Only the fixing of disclosure of the list of assertions not met as **a condition of validity of a conformance claim** is a provision of this specification
Conformance classes — C1 · C2 (11.1)	Conformance classes by role — W3C QA Framework	Specifies that the determination for each class be based on the **test records** of the per-assertion test methods (Annexes A and C — injection 36 · audit 33 · demonstration 3). Assertions designated for automated check are in principle tested by injection test (C.0.2)
Declaration of measurement provenance — measured/commanded value/default value/estimate (MEA-01)	Provenance data model — W3C PROV	A distinction specific to simulation — **separation of values produced by physics computation from input directive values**. In metrology standards, which presuppose physical measuring instruments, this distinction does not exist in principle
Standing operation of negative controls — VRF-01	Mutation testing (testing a test suite by deliberate fault injection), negative controls in clinical trials	**Candidate for codification of a paired positive control** — current = precondition of the test method (C.4.1) · codification at clause level = candidate for formal revision (12.6 ⑤). Distinguishes “a scorer that fails everything” from “a scorer that catches properly”. With negative controls alone, the two cannot be distinguished
Model authenticity — VRF-05	Tool qualification — DO-178C, ISO 26262 series	Moves the object to the authenticity of the SUT's **model specifications**. The basis is the measured observation that specifications in public repositories and specifications confirmed by the manufacturer reverse the verdict to its exact opposite

Uncertainty · decision rules	Measurement uncertainty and decision rules for statements of conformity — ISO/IEC 17025 series	**Not yet present.** This edition has no clause specifying the direction of disposition of the threshold ± uncertainty interval; it is a candidate for revision
Risk proportionality of model credibility	Credibility requirements in proportion to decision influence — ASME V&V 40, NASA-STD-7009	**Not yet present.** This specification applies the 72 assertions uniformly regardless of intended use, and the introduction of risk grades is a candidate for revision (subject to re-examination together with the prohibition of relaxation in 11.4)
Test process	Risk-based AI test process — ISO/IEC TS 42119-2:2025, ISO/IEC/IEEE 29119	The layer differs. The predecessor specifies the kinds and order of tests, and this specification specifies the basis for trusting the values those tests produce

> NOTE — The ‘Uncertainty · decision rules’ row and the ‘Risk proportionality of model credibility’ row of the table are **what this specification does not yet have**. Where this specification leaves unspecified a matter already specified by a predecessor standard, it states that fact and registers the matter as a candidate for revision.

*PAVS Part 1 Specification **v1.5** · 2026-10-20 · 72 test assertions = ITM 10 · MEA 8 · VDT 8 · VRF 16 · REP 15 · OPS 10 · CNF 5 (VRF-17 withdrawn number — see 8.17) (normative composition: shall 65 · should 6 · shall not 1 · may 0 — the authoritative count is the normative column of Annex A, **★core assertions 8** — the authoritative list is 11.2.1 · 72 test methods: injection 36 · audit 33 · demonstration 3).*