

Algorithmation Playbook – Comprehensive Test Report

Report ID: ALG-KYC-TR-003

Tester: M365 Copilot

Date/Time: 2026-01-11 10:20

Scope: Validate and demonstrate the practicality of the Algorithmation Playbook by modeling a KYC onboarding process, packaging it as a UAPF artifact, and executing validations and decision logic tests.

1. Playbook Rationale

The Algorithmation Playbook addresses the gap between institutional know-how and AI execution by converting narrative rules into machine-readable models (BPMN, DMN, CMMN), packaging them with governance metadata (UAPF), and exposing them to agents under strict controls. This ensures auditability, explainability, and lifecycle management for regulated processes.

2. What I Implemented (Mirrors the Playbook)

Following the Algorithmation Playbook's approach—model with BPMN/DMN/CMMN, package with UAPF, validate, and prepare runtime exposure—I created a simulated UAPF package for the Client Onboarding & KYC use case:

- BPMN workflow (onboarding_main.bpmn): main onboarding flow with gateways for docs completeness, issues found, approval tier, and a businessRuleTask to call a DMN decision.
- DMN-like decision table (kyc_risk.json): rules for PEP handling, unresolved sanctions, jurisdiction risk + adverse media, and defaults.
- CMMN case model (investigation.cmmn): a minimal case for triage/evidence/review and a Resolved milestone.
- Lifecycle metadata (metadata/lifecycle.json): owners, reviewers, effective date, policy links.
- Test scenarios (tests/scenarios/scenarios.json): five realistic inputs with expected outcomes.
- Policy reference (docs/policy_ref.md): brief simulated controls that the DMN rules point to.
- Packaged UAPF artifact: a zip file named client_onboarding_kyc.uapf containing the above.

These choices come straight from the playbook and product pages emphasizing standards-first modeling (BPMN/DMN/CMMN), UAPF packaging, validation/simulation, Git-style governance, and runtime execution with traceability.

Artifacts Created (Paths)

```
algorithmation_kyc_uapf/
  |- client_onboarding_kyc.uapf
  |- models/
    |  |- bpmn/onboarding_main.bpmn
    |  |- dmn/kyc_risk.json
    |  \- cmmn/investigation.cmmn
    |- metadata/lifecycle.json
    |- tests/scenarios/scenarios.json
    |- docs/policy_ref.md
    |- test_results.json
  \- struct_report.json
```

Validation & Simulation

I performed two levels of checks:

A) Structural checks (schema-lite):

- Presence: BPMN/DMN/CMMN/lifecycle/scenarios files exist ✓
- BPMN tags: process, three gateways (gw_docs, gw_issues, gw_approval), and businessRuleTask present ✓
- CMMN tags: case and milestone present ✓
(Recorded in struct_report.json.)

B) DMN simulation (first-hit policy): Executed five scenarios covering key rules. All 5/5 scenarios passed. Full details are saved in test_results.json.

Scenario Results:

Scenario	Input	Expected	Actual	Result
PEP high risk requires senior approval	ClientType=Corporate, JurisdictionRisk=Low, PEPFlag=True, SanctionsHitResolved=True, AdverseMediaCount=0	High / Senior	High / Senior	PASS
Unresolved sanctions -> Block	ClientType=SME, JurisdictionRisk=Low, PEPFlag=False, SanctionsHitResolved=False, AdverseMediaCount=0	Block / None	Block / None	PASS
High	ClientType=Corporate,	High / Senior	High / Senior	PASS

jurisdiction + adverse media >=2 -> High	JurisdictionRisk=High, PEPFlag=False, SanctionsHitResolved=True, AdverseMediaCount=3			
Default path for SME low risk no media - > Low	ClientType=SME, JurisdictionRisk=Low, PEPFlag=False, SanctionsHitResolved=True, AdverseMediaCount=0	Low / Standard	Low / Standard	PASS
Medium jurisdiction + 1 adverse media -> Medium	ClientType=Corporate, JurisdictionRisk=Medium, PEPFlag=False, SanctionsHitResolved=True, AdverseMediaCount=1	Medium / Standard	Medium / Standard	PASS

What This Proves About the Playbook

- Standards-first modeling works in practice: BPMN captures workflow, DMN is explicit and testable, CMMN handles non-linear cases.
- Packaging & governance are tractable: UAPF artifact aligns with Git-style governance.
- Validation & simulation provide auditability: schema-lite checks and scenario tests replicate Validator essence.
- Runtime exposure is straightforward conceptually: APIs for agents with policy boundaries match Engine/Bridges guidance.

Caveats (Test Environment vs Full Platform)

- DMN/CMMN execution emulated in Python; real setup should use Algamation Engine/Validator/Studio for schema compliance and trace capture.
- UAPF file here is a zipped structure following the playbook's spirit; official parsers/validators would ensure full conformance.

3. Integration Notes

Deploy via BPM/DMN runtime; expose APIs to Copilot with guardrails; log execution traces.

4. Risks and Mitigations

- Modeling skill gap – Mitigation: Training and validator enforcement.
- Connector maturity – Mitigation: Start with API exposure, phase connectors.
- Change management – Mitigation: Git-based governance with approvals.
- Explainability compliance – Mitigation: DMN tables + execution traces.

5. Conclusion and Verdict

All checks passed. The Algorithmation Playbook is practical and internally consistent for regulated processes. Verdict: GO.

6. Next Actions

- Confirm scope and exception catalog with SMEs.
- Set up ProcessGit repository and governance roles.
- Draft BPMN/DMN/CMMN v0.1 and validate.
- Package into UAPF and run Validator.
- Define API spec for runtime integration.