KRAFTHAUS RESEARCH PAPER

Decision Packets for KPI-Critical Product Decisions: A Design Science Artifact and Evaluation Framework

Abstract

Organizations repeatedly make KPI-critical product and growth decisions under time pressure, fragmented evidence, and cross-functional disagreement. Despite mature analytics and experimentation stacks, many teams still fail to produce one executable recommendation tied to one owner, one dispatch window, and one auditable rationale. This paper presents a design science artifact, instantiated in Krafthaus, that operationalizes recurring decisions as structured Decision Packets. The artifact enforces a minimum context contract, bounded option readiness, reliability controls, and dual outputs for execution and governance (machine-readable packet and leadership-readable report). We define design objectives and principles, specify a platform-supporting architecture, and provide testable propositions with a longitudinal evaluation protocol. The paper contributes: (1) a transferable operating model for decision closure under bounded rationality; (2) a stable decision contract that decouples human framing from runtime execution details; and (3) an implementation pathway from ad hoc recommendation work to cumulative, auditable organizational learning.

Keywords: decision support systems, design science research, product decision making, KPI governance, decision operations, auditability, reliability engineering.

1. Introduction

Product, growth, and operations teams make high-frequency decisions that are strategically meaningful yet operationally under-specified. Typical examples include pricing anchor choices, onboarding gate policies, and feature-priority tradeoffs. These decisions are often delayed not because evidence is unavailable, but because organizations lack a repeatable mechanism to transform deliberation into one executable call.

This gap is consistent with bounded rationality: under finite attention and time, organizations satisfice through social negotiation rather than optimized, traceable closure (Simon, 1955). In high-velocity settings, process structure and conflict discipline are strongly associated with superior decision speed and quality (Eisenhardt, 1989).

Current tool categories address parts of this problem but rarely the full closure loop. Analytics systems improve measurement, experimentation systems improve variant inference, and policy engines improve deterministic enforcement. However, few systems guarantee that a bounded business question is consistently converted into an accountable recommendation, execution handoff, and auditable record.

This paper addresses that gap through a design science artifact and asks:

Research Question (RQ): How can organizations implement a low-overhead, platform-supporting mechanism that repeatedly converts KPI-critical deliberation into executable and auditable decision closure?

The paper makes three contributions:

1. A constrained decision-unit model and minimum context contract for operational decisions.
2. A platform-supporting Decision-Packet architecture with explicit reliability and governance controls.
3. A proposition-driven evaluation framework linking process design to operational and KPI outcomes.

2. Theoretical Foundation

2.1 Bounded rationality and closure failure

Bounded rationality explains why organizational decisions often remain under-closed under practical constraints (Simon, 1955). In sprint-based environments, three closure failures recur:

- scope diffusion (question broadening),
- option instability (drifting alternatives),
- rationale decay (handoff without durable justification).

2.2 Process structure in high-velocity decisions

Research on fast strategic decisions indicates that effective organizations often rely on disciplined process and rapid closure mechanisms rather than maximal analytical breadth (Eisenhardt, 1989). This motivates system designs that optimize closure mechanics and accountability.

2.3 DSS and design science relevance

Decision support systems integrate data, models, and managerial judgment (Keen & Scott Morton, 1978; Power, 2002). Yet many enterprise DSS approaches impose integration overhead that is poorly matched to recurring weekly/sprint decisions. Design science is therefore appropriate: build and evaluate an artifact for an important class of practical problems (Hevner et al., 2004; Peffers et al., 2007; Gregor & Hevner, 2013).

2.4 Governance constraints for AI-assisted decisions

Trustworthy AI-assisted decisions require transparency, accountability, and risk controls. Governance frameworks prioritize controllability, traceability, and human oversight over unconstrained automated advice (NIST AI RMF 1.0).

3. Research Method

3.1 Design science process

The study follows a DSR sequence:

1. Problem identification: recurring KPI decisions fail to close reliably.
2. Objective definition: low-friction intake, executable recommendation, resilience, and auditability.
3. Design/development: Decision-Packet architecture and operator control plane.
4. Demonstration: live workflow with queue and lifecycle states.
5. Evaluation design: utility, reliability, and governance metrics.
6. Communication: this manuscript and companion technical documentation.

3.2 Scope of claims

This paper establishes artifact design validity and evaluability for the target decision class. Causal performance claims are defined as empirical hypotheses and are evaluated through the protocol in Section 8.

4. Problem Class and Design Objectives

4.1 Problem class definition

The target class comprises decisions that are:

- recurring (weekly/sprint cadence),
- KPI-linked,
- cross-functional,
- execution-sensitive,
- bounded enough to compare finite alternatives.

4.2 Design objectives

The artifact must satisfy six objectives:

O1: Minimum context sufficiency Collect only context needed for reliable recommendation quality.

O2: Actionable closure Produce one clear call tied to one owner and one dispatch window.

O3: Option comparability Maintain fixed option boundaries and an explicit ranking basis.

O4: Operational resilience Remain usable during upstream degradation and transient failures.

O5: Auditability Preserve traceable rationale, request lineage, and decision metadata.

O6: Learning loop support Capture outcomes for threshold, process, and calibration improvement.

4.3 Minimum context sufficiency proposition

A free-text decision question is generally insufficient for reliable operational recommendation. The minimum useful context includes:

- KPI metric (and optional baseline/target),
- bounded option set,
- explicit constraints,
- accountable owner,
- dispatch/evaluation window.

This principle balances recommendation quality against adoption friction.

5. Artifact Instantiation: Krafthaus Decision-Packet System

5.1 Decision unit

A decision unit is defined as:

- one KPI-linked question,
- one accountable owner,
- two to six options,
- explicit constraints,

- one dispatch window.

This unit intentionally excludes broad strategic ideation and open-ended exploratory research.

5.2 Intake and contract formation

Structured intake transforms operator submissions into a normalized context contract. Validation gates enforce:

- required context checks,
- readiness-dependent option rules,
- format and length constraints,
- anti-abuse controls.

5.3 Decision engine abstraction

The frontend calls an internal endpoint (/api/decision-evaluate) rather than direct upstream providers. This boundary enables:

- API key isolation,
- provider abstraction,
- response normalization,
- uniform observability.

5.4 Reliability controls

Production operation includes:

- bounded timeout,
- retry on timeout/5xx,
- circuit breaker (threshold/window/cooldown),
- graceful degraded fallback preserving response shape.

These controls reduce operator-blocking incidents during upstream instability.

5.5 Decision packet and report artifacts

Two synchronized outputs are emitted:

1. Machine-readable Decision Packet for integration, traceability, and lifecycle governance.
2. Leadership-readable report for execution alignment and operational communication.

5.6 Lifecycle control plane

Explicit lifecycle states include:

- submitted,
- scope confirmed,
- payment confirmed,
- in execution,
- delivered,
- closed.

This state model reduces hidden coordination load and supports accountable handoff.

6. Design Principles

DP1: Constrain before compute Reliable recommendations require bounded decision context before runtime evaluation.

DP2: Normalize before render Stable contracts should precede all UI and report generation.

DP3: Degrade safely Failures should return usable, explicitly degraded outputs rather than hard process breaks.

DP4: Trace by default Every recommendation should include lineage and rationale metadata.

DP5: Close and learn Decision systems should capture post-decision outcomes for recalibration.

7. Testable Propositions

P1 (Cycle-Time Reduction): Packetized decision workflows will show lower median decision cycle time than matched non-packet workflows.

P2 (Actionability Increase): Packetized decision workflows will produce higher same-window action rates.

P3 (Governance Improvement): Packetized decision workflows will reduce rationale-loss incidents at implementation handoff.

P4 (Reliability Benefit): Retry + circuit breaker + fallback controls will reduce operator-blocking incidents during upstream instability.

P5 (Learning Effect): Repeated packet closure with outcome capture will reduce recommendation reversal rates over time.

8. Evaluation Framework

8.1 Study design

The evaluation uses a 12-24 week quasi-experiment with either:

- pre/post adoption within teams, or
- matched treatment/control cohorts.

8.2 Unit of analysis

Decision packet (or matched non-packet decision event).

8.3 Outcome measures

Primary outcomes

- decision cycle time,
- same-window action rate,
- reversal rate,
- override rate (reason-coded),
- KPI delta over declared window.

Operational outcomes

- started/ok/degraded/failed event rates,
- latency p50/p95,
- circuit-breaker open frequency,
- fallback invocation frequency.

8.4 Analysis plan

- survival analysis for cycle-time outcomes,
- logistic regression for actionability/reversal,
- mixed-effects models for KPI deltas across teams and decision types.

8.5 Validity controls

Control for decision type, risk tier, option readiness, owner seniority, and release-season effects.

9. Discussion

9.1 Theoretical implications

The artifact defines a middle layer between traditional DSS and unconstrained AI advisory interfaces. It frames decision quality as a system property: context constraints + contract stability + reliability semantics.

9.2 Practical implications

For product organizations, the artifact reduces ambiguity-to-execution latency. For leadership, it improves traceability and governance. For platform teams, it enables runtime-provider portability via stable contracts.

9.3 Platform-supporting implications

The architecture supports layered maturity:

- MVP layer: intake -> packet/report,
- operational layer: queue + lifecycle governance,
- platform layer: calibration loops, benchmark baselines, multi-domain expansion.

10. Threats to Validity and Limitations

1. Early adopter bias may inflate observed gains.
2. Not all strategic questions are packetizable into bounded options.
3. Recommendation quality remains input-quality dependent.
4. KPI outcomes have confounders outside the decision workflow.

5. Organizational process maturity may moderate treatment effects.

11. Ethics and Governance

Required controls include:

- explicit human accountability for final calls,
- transparent confidence/tradeoff reporting,
- data minimization,
- reason-coded overrides,
- periodic calibration and bias review.

These controls are necessary for trustworthy deployment of operational decision systems.

12. Conclusion

This paper presents a design science artifact for operational decision closure in KPI-critical product contexts. By combining minimum context sufficiency, contract normalization, reliability controls, and auditable outputs, the architecture provides a low-overhead and governable mechanism for recurring decisions. The design is platform-supporting and transferable across runtime providers and domains. The evaluation protocol defines how cycle time, actionability, reversals, and KPI movement should be measured and compared in production deployments.

References

- Eisenhardt, K. M. (1989). Making fast strategic decisions in high-velocity environments. Academy of Management Journal, 32(3), 543-576. https://doi.org/10.2307/256434
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. MIS Quarterly, 37(2), 337-355. https://doi.org/10.25300/MISQ/2013/37.2.01
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75-105. https://doi.org/10.2307/25148625
- Keen, P. G. W., & Scott Morton, M. S. (1978). Decision Support Systems: An Organizational Perspective. Addison-Wesley.
- NIST. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

- Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45-77. https://doi.org/10.2753/MIS0742-1222240302
- Power, D. J. (2002). Decision Support Systems: Concepts and Resources for Managers. Quorum Books.
- Simon, H. A. (1955). A behavioral model of rational choice. Quarterly Journal of Economics, 69(1), 99-118. https://doi.org/10.2307/1884852