Krafthaus and the Decision Packet: A Design-Science Framework for KPI-Critical Product Decisions

A formal design-science account of how Krafthaus converts ambiguous product deliberation into accountable decision packets, decision memos, and audit traces.

MANUSCRIPT TYPE

Public design-science thesis and evaluation framework

VERSION

Public thesis, May 2026

PREPARED BY

Krafthaus

Contents

Abstract

Product, growth, and operations teams rarely fail because they lack another dashboard. They fail because ambiguous deliberation is not reliably converted into one accountable decision, tied to one metric, one owner, one execution window, and one auditable rationale. This paper presents the Krafthaus thesis: recurring KPI-critical decisions should be treated as structured decision packets rather than as meetings, slide decks, or unconstrained advisory prompts. A decision packet is a bounded artifact containing a question, a finite option set, a KPI, guardrails, evidence, ownership, and lifecycle metadata. Krafthaus instantiates this artifact as a workflow that turns a submitted decision question into a confirmed packet and then into a decision memo with an audit trace.

The paper contributes a formal model, a platform architecture, and an evaluation protocol. Formally, we define a decision packet as $D = \langle q, O, k, G, E, \tau, \omega, \varrho \rangle$, define option feasibility and ranking functions, and separate decision edge from quantitative impact forecasts. Architecturally, we position Decision API as the validation and runtime layer beneath Krafthaus: it normalizes packet state, evaluates readiness, preserves deterministic contracts where appropriate, and supports traceable execution. Evaluation is specified as a longitudinal design-science program, not as a completed causal field claim. The proposed measures include cycle time, same-window action rate, reversal rate, guardrail violations, audit completeness, and post-decision KPI movement when a backed measurement model exists.

The central claim is intentionally precise: Krafthaus does not eliminate judgment; it operationalizes judgment into a repeatable, inspectable, and learnable unit of work.

Keywords: decision support systems, design science research, decision operations, product strategy, KPI governance, auditability, policy runtime, bounded rationality.

---

1. Introduction

Modern product organizations are surrounded by analytic tools, experimentation platforms, research repositories, and collaboration software. Yet many consequential decisions still end in ambiguity. A team may know the dashboard, understand the customer feedback, and still fail to produce one executable recommendation. The failure is often not informational; it is procedural. The organization lacks a stable mechanism for turning deliberation into closure.

Krafthaus addresses this gap through the Decision Packet. The packet is a constrained unit of decision work. It asks a scoped question, confirms the candidate options, binds the decision to a KPI and guardrails, names an accountable owner, and emits a memo with rationale and audit trace. The product expression is deliberately simple: one decision credit buys one decision run; the user submits the question and required context; Krafthaus confirms the packet before execution; the output is a decision memo and audit trace.

This paper formalizes the thesis behind that product. It asks:

Research question. How can organizations implement a low-overhead mechanism that repeatedly converts KPI-critical deliberation into executable and auditable decision closure?

The answer proposed here is not "more AI" in the generic sense. It is a design-science artifact: a structured decision packet, coupled to runtime validation and a memo output contract. Krafthaus may use model-based analysis, deterministic validation, and software automation, but the core artifact is organizational rather than merely computational. Its purpose is to reduce the distance between "we discussed this" and "we made the call, know why, and know what to do next."

The paper makes four contributions:

1. It defines the problem class of recurring KPI-critical decisions.
2. It specifies a formal packet model, including feasibility, scoring, decision edge, and audit trace notation.
3. It describes Krafthaus as an instantiated decision-packet system, with Decision API as the validation and runtime substrate.
4. It provides a conservative evaluation protocol that separates propositions from causal claims.

---

2. Theoretical Background

2.1 Bounded rationality and closure failure

Bounded rationality explains why organizations rarely optimize decisions under ideal conditions. Decision makers operate under limited time, attention, and computational capacity (Simon, 1955). In product teams, this manifests as scope drift, option churn, and rationale decay. A question begins as "which pricing rule should we ship?" and becomes a discussion about segmentation, launch timing, competitor movement, customer success load, and brand positioning. Each consideration may be valid, but without a closure mechanism the decision object dissolves.

The Decision Packet treats closure as a design requirement. It does not assume that teams can or should process all available information. Instead, it creates a bounded context in which information is sufficient for the next executable decision.

2.2 High-velocity decision process

Research on high-velocity strategic decisions shows that speed and quality can improve when teams use disciplined process rather than unbounded analysis (Eisenhardt, 1989). The relevant lesson for Krafthaus is not that decisions should be rushed. It is that decision quality depends on the structure of comparison, conflict resolution, and accountability. A system that forces option comparability, guardrail articulation, and owner assignment can improve the conditions under which judgment operates.

2.3 Decision support systems and design science

Decision support systems combine data, models, and managerial judgment (Keen & Scott Morton, 1978; Power, 2002). Traditional DSS architectures often assume that the decision object has already been defined. In fast-cycle product work, that assumption is too strong. The upstream problem is frequently the formation of the decision object itself.

Design science research is appropriate because the contribution is artifact-centered: build, specify, and evaluate a system that addresses a practical organizational problem (Hevner et al., 2004; Peffers et al., 2007; Gregor & Hevner, 2013). Krafthaus is therefore framed as an artifact with design objectives and evaluable propositions, not as a claim that one universal model can choose correctly in all domains.

2.4 Governance for AI-assisted judgment

AI-assisted decision workflows introduce risks: false precision, untraceable rationale, overreliance, and unreviewed automation. Trustworthy deployment requires explicit boundaries, human accountability, provenance, and monitoring. Krafthaus follows a control-first posture: the system returns a memo and audit trace; it does not erase the owner of the decision. Forecasts are shown only when backed by an explicit measurement or modeling contract. Otherwise, the system may rank options by score, confidence, guardrail fit, and decision edge without pretending that a dollar estimate exists.

---

3. Problem Class and Scope Conditions

The target problem class is the recurring KPI-critical product decision. These decisions have five properties:

1. **Bounded consequence.** The decision matters, but it is narrow enough to act on within a defined window.
2. **Finite options.** There are at least two candidate options, and the set can be confirmed before execution.
3. **KPI relevance.** The decision is tied to a metric such as activation, expansion, churn, support load, latency, retention, or conversion.
4. **Guardrail dependence.** Some options may be unacceptable even if they improve the focal KPI.
5. **Handoff sensitivity.** The decision must be communicated to people or systems that need rationale, ownership, and next steps.

Examples include pricing-packaging choices, onboarding gate policies, usage-limit decisions, launch sequencing, escalation rules, lifecycle messaging, and feature-priority tradeoffs. The class excludes unbounded strategy formation, purely creative ideation, and decisions where the ethical or legal responsibility cannot remain with an accountable human owner.

4. Formal Model

4.1 Decision packet

Let a decision packet be:

$$D = \langle q, O, k, G, E, \tau, \omega, \varrho \rangle$$

where:

- q is the decision question.
- $O = \{o_1, \dots, o_n\}$ is the confirmed option set, with $n \geq 2$.
- k is the focal KPI or outcome metric.
- $G = \{g_1, \dots, g_m\}$ is the guardrail set.
- E is the evidence bundle.
- τ is the decision and evaluation time window.
- ω is the accountable owner.
- ϱ is lifecycle and provenance metadata.

The packet is valid only when a minimum context predicate is satisfied:

$$\chi(D) = \mathbf{1}[q \neq \varnothing] \cdot \mathbf{1}[|O| \geq 2] \cdot \mathbf{1}[k \neq \varnothing] \cdot \mathbf{1}[\omega \neq \varnothing] \cdot \mathbf{1}[\tau \neq \varnothing].$$

If $\chi(D) = 0$, Krafthaus should request repair rather than fabricate precision.

4.2 Feasibility

Each guardrail g_j defines an admissible boundary b_j and a measurement direction. Let $v_j(o_i)$ be the estimated guardrail value for option o_i. Feasibility is:

$$F(o_i \mid D) = \prod_{j=1}^{m} \mathbf{1}[v_j(o_i) \leq b_j],$$

with the inequality reversed where the guardrail is minimum-threshold rather than maximum-threshold. Options with $F(o_i \mid D) = 0$ may still be discussed, but they cannot be recommended without an explicit override.

4.3 Ranking score

For a feasible option, define a decision score:

$$S(o_i \mid D) = \alpha U_i + \beta C_i + \eta A_i - \gamma R_i - \lambda V_i - \mu H_i.$$

The terms are:

- U_i: expected utility with respect to KPI k.
- C_i: confidence in the option's evidence base.
- A_i: actionability or implementation readiness.
- R_i: downside risk.
- V_i: guardrail violation pressure.
- H_i: handoff complexity.
- $\alpha, \beta, \eta, \gamma, \lambda, \mu \geq 0$: weights determined by packet type and governance policy.

The recommended option is:

$$o^{*} = \arg \max_{o_i \in O : F(o_i \mid D) = 1} S(o_i \mid D).$$

4.4 Decision edge

The decision edge is the separation between the selected option and the next-best feasible option:

$$\Delta(D) = S(o^{*} \mid D) - \max_{o_i \in O,\, o_i \neq o^{*},\, F(o_i \mid D) = 1} S(o_i \mid D).$$

A low $\Delta(D)$ indicates a fragile recommendation even when a winner exists. This distinction matters commercially and ethically. Krafthaus can report a winner with rationale while also disclosing that the decision edge is narrow.

4.5 Impact forecasts

Let I_i be a quantitative forecast for option o_i. I_i is defined only if a forecast contract exists:

$$\psi_i = \mathbf{1}[\text{baseline observed}] \cdot \mathbf{1}[\text{forecast unit specified}] \cdot \mathbf{1}[\text{model or measurement source recorded}].$$

If $\psi_i = 0$, then I_i is withheld and the memo should not display an invented dollar or KPI forecast. In that case, Krafthaus may still rank options using $S(o_i \mid D)$, C_i, guardrail fit, and decision edge. This is a trust boundary: absence of a backed forecast is information, not a defect.

4.6 Closure function

The system output is a closure function:

$$
\Phi(D) = \begin{cases} (o^{\backslash *}, M, A), & \chi(D) = 1 \text{ and } \exists o_i : F(o_i \mid D) = 1, \\ \text{repair request}, & \chi(D) = 0, \\ \text{blocked memo}, & \forall o_i, F(o_i \mid D) = 0. \end{cases}
$$

where M is the decision memo and A is the audit trace.

4.7 Audit trace

The audit trace is an ordered event sequence:

$$
A = \{a_t\}_{t=1}^{T}, \quad a_t = \langle t, u_t, x_t, h_t, s_t \rangle,
$$

where t is timestamp, u_t is actor or system component, x_t is event payload, h_t is a content hash or stable identifier, and s_t is state after event application. The trace is valuable because it allows a decision to be reconstructed without relying on memory, screenshots, or meeting folklore.

---

5. Artifact Instantiation: Krafthaus

5.1 Product workflow

Krafthaus instantiates the formal model as a decision workflow:

1. A user submits one decision question, one KPI, candidate options, and guardrails.
2. Krafthaus confirms the packet before execution.
3. The system runs the packet through validation, evidence normalization, and ranking logic.
4. The user receives a decision memo with winner, rationale, next steps, and audit trace.

The public product expression is intentionally narrow: a credit purchases a decision run; if the system cannot deliver the promised artifact, operational policy may restore the credit. This commercial structure reinforces the thesis. Krafthaus is not selling generic access to a chat model; it is selling accountable decision closure.

5.2 Decision API as runtime substrate

Decision API is best understood as the lower-level validation and runtime layer beneath the Krafthaus product. It performs or supports:

- packet schema validation,
- runtime configuration and readiness checks,
- normalized response contracts,
- deterministic policy or scoring paths where applicable,

- provider abstraction,
- observability and degradation handling.

In architectural notation:

$$\text{Krafthaus} = \text{UX}_{intake} \circ \text{Packet}_{contract} \circ \text{DecisionAPI}_{runtime} \circ \text{Memo}_{render} \circ \text{Audit}_{trace}.$$

This composition preserves a useful separation. Krafthaus is the organizational artifact and customer-facing workflow. Decision API is the execution substrate that makes the artifact stable enough to operate.

5.3 Lifecycle states

A decision run moves through explicit lifecycle states:

$$L = \{\text{submitted}, \text{packet-confirmed}, \text{payment-confirmed}, \text{running}, \text{delivered}, \text{closed}\}.$$

Explicit lifecycle states reduce hidden coordination cost. They also make the system observable: an undelivered run is not a vague failure; it is a state transition problem with a trace.

5.4 Memo output contract

The memo is not a slide deck and not a transcript. It is a closure artifact. A trustworthy memo includes:

- the objective question,
- the confirmed options,
- the selected option,
- rationale for selection,
- why alternatives were not selected,
- guardrail review,
- next steps,
- confidence and limitations,
- audit trace or audit reference.

The memo may include quantitative impact only when the forecast contract in Section 4.5 is satisfied. Otherwise, the memo should report decision edge, confidence, and guardrail reasoning without pretending to know a precise financial result.

5.5 Reliability controls

Operational trust depends on failure semantics. A decision system should not silently collapse when upstream dependencies degrade. Krafthaus therefore requires:

- bounded timeouts,
- retries for transient failure,
- circuit breakers for repeated upstream instability,
- degraded but shape-preserving responses when possible,
- explicit blocked states when repair is required.

The user should always be able to distinguish "the system made a backed claim" from "the system preserved workflow continuity under degraded conditions."

---

6. Design Principles

DP1: Constrain before compute. The system should first make the decision object bounded. Computation over an unstable object creates false confidence.

DP2: Confirm options before ranking. Candidate options should be confirmed as a set before the output declares a winner.

DP3: Separate ranking from forecasting. A strong ranking can exist without a backed dollar forecast. The product should not conflate these.

DP4: Preserve human accountability. Krafthaus supports a call; it does not erase the owner of that call.

DP5: Render for action, not admiration. The output should tell the operator what to do next and why, rather than showcase analysis for its own sake.

DP6: Trace by default. If the rationale cannot be reconstructed later, the system has not delivered a durable decision artifact.

DP7: Learn only from recorded outcomes. Calibration should be based on outcome capture, not anecdotal satisfaction.

---

7. Propositions

The following propositions are designed for empirical evaluation. They are not claimed as already proven.

P1: Cycle-time compression. For matched KPI-critical decisions, packetized workflows will reduce median time from question submission to executable recommendation.

P2: Actionability improvement. Packetized workflows will increase the share of decisions acted upon within the declared dispatch window.

P3: Rationale preservation. Packetized workflows will reduce rationale-loss incidents during handoff, measured by whether implementers can reconstruct why the selected option won.

P4: Guardrail discipline. Explicit guardrail specification will reduce post-decision violations relative to unstructured recommendation workflows.

P5: Forecast honesty. Withholding unbacked forecasts will increase perceived trustworthiness relative to systems that display unsupported numerical impact estimates.

P6: Learning-loop benefit. Repeated packet closure with outcome capture will reduce reversal rates and improve calibration over time.

---

8. Evaluation Protocol

8.1 Unit of analysis

The unit of analysis is the decision event. For Krafthaus, this is a delivered decision packet and memo. For comparison workflows, it is the matched meeting, document, or issue thread that resulted in a decision.

8.2 Study design

A practical evaluation can use one of three designs:

1. **Pre/post adoption.** Compare teams before and after Krafthaus adoption.
2. **Matched cohort.** Compare Krafthaus-supported decisions to similar decisions handled without the packet workflow.
3. **Within-team randomized routing.** Route eligible decisions to packetized or non-packetized workflow where operationally acceptable.

The third design has the strongest identification but may be impractical for early customers. The first two are still useful if confounders are recorded.

8.3 Primary measures

Let T_c be cycle time, A_w same-window action, R_v reversal, G_v guardrail violation, and Q_m memo completeness.

$$Y = \{T_c, A_w, R_v, G_v, Q_m, \Delta k\}.$$

The KPI movement Δk is included only when measurement attribution is defensible:

$$\Delta k = k_{post,\tau} - k_{pre}.$$

If attribution is weak, Δk should be reported as descriptive, not causal.

8.4 Statistical analysis

Recommended analysis includes:

- survival analysis for cycle-time outcomes,
- logistic regression for actionability and reversal,
- mixed-effects models for repeated team-level observations,
- sensitivity analysis by decision type, risk tier, and option count.

A simple mixed-effects specification is:

$$Y_{d,t} = \theta_0 + \theta_1 Packetized_{d,t} + \theta_2 RiskTier_{d,t} + \theta_3 OptionCount_{d,t} + u_{team} + \epsilon_{d,t}.$$

The coefficient θ_1 estimates the association between packetization and outcome after controls. Causal interpretation requires stronger design assumptions.

8.5 Qualitative review

Because decision quality is partly interpretive, evaluation should include structured qualitative review. Reviewers should rate:

- whether the question was scoped,
- whether options were comparable,
- whether rationale was specific,
- whether guardrails were acknowledged,
- whether the next action was clear,
- whether uncertainty was honestly disclosed.

These ratings produce an audit-completeness score:

$$Q_m = \frac{1}{r} \sum_{\ell=1}^{r} z_\ell, \quad z_\ell \in \{0, 1\}.$$

---

9. Trust Boundaries

The thesis is strongest when its boundaries are explicit.

First, Krafthaus is not a replacement for executive accountability. A memo can recommend; the owner remains responsible for accepting, rejecting, or overriding the recommendation.

Second, Krafthaus does not guarantee KPI improvement from every recommendation. It creates a better decision artifact and a more evaluable process. KPI movement remains subject to market conditions, implementation quality, and measurement noise.

Third, Krafthaus should not display quantitative impact estimates unless the estimate is backed by a recorded baseline, forecast unit, and source. Decision edge and confidence are valid outputs; fabricated precision is not.

Fourth, Krafthaus is not a legal, medical, or investment authority. It is intended for bounded product, growth, and operational decisions where accountable business owners can review the output.

These boundaries are not marketing weaknesses. They are the basis of durable trust.

---

10. Discussion

10.1 Theoretical implication

Krafthaus reframes decision quality as an artifact property. The quality of a decision is not only a function of the model or the decision maker; it is also a function of how the decision object is formed, constrained, rendered, and preserved. This adds a practical layer between classical DSS and contemporary AI copilots.

10.2 Practical implication

For teams, the packet reduces the coordination cost of reaching an executable call. For leaders, the memo reduces ambiguity around why a decision was made. For platform operators, the runtime contract creates a stable point for validation, monitoring, and future integration.

10.3 Product implication

The credit-based Krafthaus model aligns payment with delivered closure rather than seat usage or token consumption. The customer pays for the decision artifact: packet confirmation, memo, and audit trace. This commercial design is consistent with the thesis because it centers the unit of value on decision closure.

---

11. Limitations

This paper has five limitations.

First, it formalizes an artifact and evaluation protocol rather than reporting completed longitudinal causal estimates.

Second, not all decisions are packetizable. Some strategic questions require discovery before options can be bounded.

Third, recommendation quality depends on input quality. A poorly scoped question with weak evidence can still produce a weak decision.

Fourth, formal notation may imply more determinism than the organizational setting allows. The notation is intended to make assumptions inspectable, not to pretend that judgment has been eliminated.

Fifth, early adopters may have unusually high process discipline, which can bias evaluation results unless controlled.

12. Conclusion

The Krafthaus thesis is that KPI-critical decisions should become durable artifacts. A decision should not disappear into a meeting note, a Slack thread, or a model response. It should be framed as a packet, checked for context, evaluated against options and guardrails, rendered as a memo, and preserved through an audit trace.

This approach does not remove uncertainty. It makes uncertainty legible. It does not replace judgment. It gives judgment a repeatable operating form. In that sense, Krafthaus is best understood as decision infrastructure: a practical system for converting ambiguous deliberation into accountable, inspectable, and learnable decision closure.

References

- Cyert, R. M., & March, J. G. (1963). *A Behavioral Theory of the Firm*. Prentice-Hall.
- Davenport, T. H., & Harris, J. G. (2007). *Competing on Analytics: The New Science of Winning*. Harvard Business School Press.
- Eisenhardt, K. M. (1989). Making fast strategic decisions in high-velocity environments. *Academy of Management Journal, 32*(3), 543-576.
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2), 337-355.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.
- Kahneman, D., & Klein, G. (2009). Conditions for intuitive expertise: A failure to disagree. *American Psychologist, 64*(6), 515-526.
- Keen, P. G. W., & Scott Morton, M. S. (1978). *Decision Support Systems: An Organizational Perspective*. Addison-Wesley.
- March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science, 2*(1), 71-87.
- National Institute of Standards and Technology. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. 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.
- 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.
- World Wide Web Consortium. (2013). *PROV-DM: The PROV Data Model*. W3C Recommendation.