

# 10x

## Outcome-Based Agile in the Era of Human–AI Software Development

**Michel Blomgren**

*Software Engineer & Solution Architect*

[s6mwa@gmail.com](mailto:s6mwa@gmail.com) — [pkt.systems](http://pkt.systems)

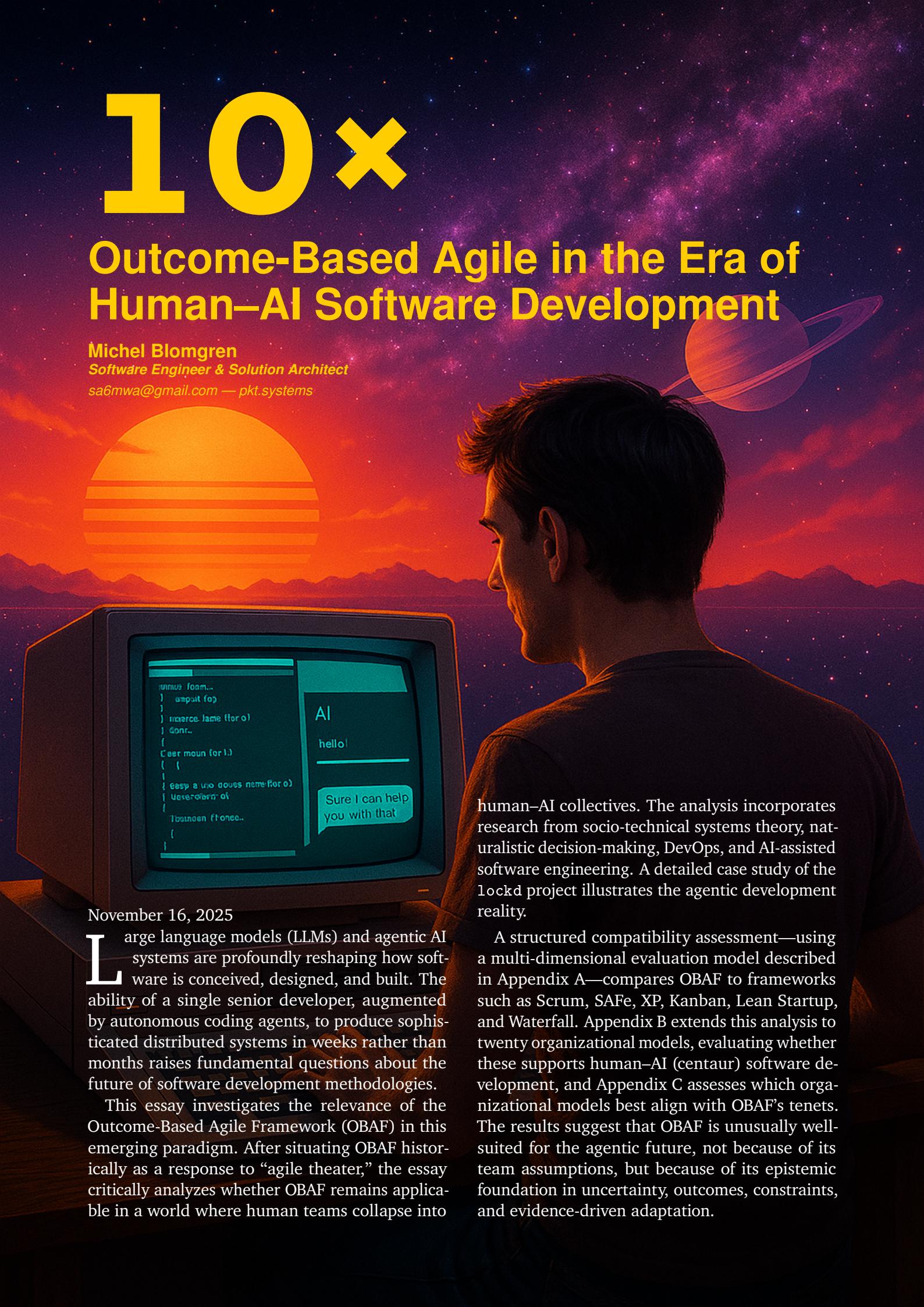
November 16, 2025

**L**arge language models (LLMs) and agentic AI systems are profoundly reshaping how software is conceived, designed, and built. The ability of a single senior developer, augmented by autonomous coding agents, to produce sophisticated distributed systems in weeks rather than months raises fundamental questions about the future of software development methodologies.

This essay investigates the relevance of the Outcome-Based Agile Framework (OBAF) in this emerging paradigm. After situating OBAF historically as a response to “agile theater,” the essay critically analyzes whether OBAF remains applicable in a world where human teams collapse into

human–AI collectives. The analysis incorporates research from socio-technical systems theory, naturalistic decision-making, DevOps, and AI-assisted software engineering. A detailed case study of the *lockd* project illustrates the agentic development reality.

A structured compatibility assessment—using a multi-dimensional evaluation model described in Appendix A—compares OBAF to frameworks such as Scrum, SAFe, XP, Kanban, Lean Startup, and Waterfall. Appendix B extends this analysis to twenty organizational models, evaluating whether these supports human–AI (centaur) software development, and Appendix C assesses which organizational models best align with OBAF’s tenets. The results suggest that OBAF is unusually well-suited for the agentic future, not because of its team assumptions, but because of its epistemic foundation in uncertainty, outcomes, constraints, and evidence-driven adaptation.



## INTRODUCTION

Software engineering is undergoing a methodological, organizational, and cognitive shift without historical precedent. The emergence of advanced large language models (LLMs), autonomous coding assistants, and multi-agent AI systems is not simply increasing productivity; it is changing the very substrate of how software is reasoned about, created, and maintained. Assumptions that have governed decades of engineering practice—including team-based coordination, incremental design elaboration, and the economics of iteration—are now **destabilized** by the presence of agentic systems that collaborate, generate, refactor, explore, and synthesize at machine speed.

The `lockd` project is emblematic of this transition (Blomgren, 2025a). Developed in roughly five weeks of spare time by a single senior engineer working in collaboration with an AI coding assistant, `lockd` evolved rapidly into a coherent distributed system with capabilities normally associated with multi-person platform teams: a distributed lock service with strong consistency semantics, an embedded NoSQL document store, an indexing and query engine (with a query language – LQL), a messaging system (LQ), a multi-backend storage abstraction, and a streaming-response execution model. The project’s AGENTS .md file documents a rich human–AI interaction pattern, where the developer framed intent, constraints, and invariants, while the AI agent explored alternatives, generated implementations, and refined architecture. This is not a case of “AI writing the code”; it is a case of a centaur system, an integrated human–AI cognitive unit producing outcomes neither could have achieved alone.

The term “centaur” originates from Garry Kasparov’s investigation into human–AI partnerships after his historic matches against Deep Blue (Alves and Pereira Cipriano, 2023). Kasparov observed that the strongest chess play did not emerge from humans alone or machines alone, but from a hybrid team in which humans provided contextual awareness and strategic framing, while machines provided combinatorial depth and tactical analysis. This insight now generalizes far beyond chess. In software engineering, centaur systems allow a single individual to achieve the throughput, architectural coherence, and exploratory bandwidth of an entire team. The `lockd` project provides empirical evidence of how engineering practice

transforms when human cognition is amplified—not replaced—by autonomous agents.

This essay examines whether existing development methodologies remain viable in this emerging agentic ecosystem. In particular, it considers the Outcome-Based Agile Framework (OBAF) (Blomgren, 2025b), originally conceived as a corrective to “agile theater,” and asks whether its epistemic foundations continue to hold when discovery, design, and execution are increasingly automated. To do so, the essay integrates historical analysis, conceptual reasoning, and empirical evidence from agentic engineering practices, culminating in a structured, research-backed evaluation across three layers: delivery methodologies (Appendix A), organizational architectures (Appendix B), and the alignment between OBAF and organizational models (Appendix C).

## FROM AGILE TO AGILE THEATER

Agile emerged in the early 2000s as a counter-movement to heavyweight, plan-driven methodologies such as the Waterfall model and the V-model. The theoretical and empirical critique of those earlier models is now well established: when uncertainty is high and the cost of change is lower than the cost of prediction, plan-driven methods perform poorly (Highsmith, 2001; Beck, 1999; Cockburn, 2001). Agile methods were built on a different epistemic contract: iterate quickly, learn continuously, and adapt as reality unfolds.

Yet as Agile spread into organizations of increasing size and complexity, it accreted ceremonial overhead. Scrum, SAFe, and related frameworks evolved into prescriptive systems emphasizing compliance with rituals—sprints, standups, reviews, retrospectives, PI planning—over the epistemic agility they were supposed to foster. Empirical studies now document the prevalence of “agile theater,” a performative mimicry of Agile values without their underlying cognitive intent (Kuusinen et al., 2017; Lewallen, 2020). The structural dynamics of large organizations—hierarchy, approval chains, coordination overhead—privileged ritual over learning.

## THE ORIGINS AND PHILOSOPHY OF OBAF

The Outcome-Based Agile Framework (OBAF) was created to correct this drift. OBAF reintegrates Agile with its original epistemic roots. Rather than prescribing roles or ceremonies, it emphasizes outcomes, constraints, discovery, recognitional sense-making, and evidence (Blomgren, 2025b). These principles draw from a wide range of disciplines: Boyd’s OODA loop (Osinga, 2006), naturalistic decision-making (Klein, 1998), distributed cognition (Hutchins, 1995), sociotechnical systems theory (Trist, 1981), and DevOps/continuous delivery research (Forsgren, Humble, and Kim, 2018). OBAF is procedural only in the minimal sense; its core is epistemic. It specifies how to reason in uncertainty, not how to perform rituals.

This distinction becomes essential when evaluating methodology in the context of human–AI teaming. OBAF does not depend on team size, role boundaries, or ceremonies. Its principles survived the collapse of human-coordination constraints because they were never premised on them.

## THE AGENTIC FUTURE OF SOFTWARE DEVELOPMENT

Recent empirical work demonstrates that LLMs and autonomous agents accelerate the most cognitively demanding aspects of software engineering: code generation, test synthesis, refactoring, exploratory design, and system analysis (Chen, 2021; Bavishi, 2022). Multi-agent systems extend these capabilities into parallelized execution, autonomous experimentation, and distributed architectural reasoning (Qian et al., 2023; Zhao, 2023).

The implications are profound. Coordination costs—a dominant bottleneck in human-only software teams (Brooks, 1995)—collapse when tasks are decomposed and executed by AI agents. Discovery, historically expensive and episodic, becomes continuous and automated. Architectural exploration expands far beyond human working memory. The “team” itself becomes a hybrid cognitive entity: a centaur composed of one human orchestrator and a constellation of autonomous agents capable of executing, evaluating, and recomposing work in real time.

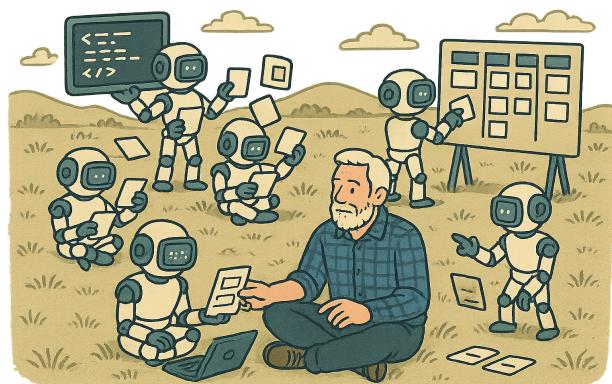
In this environment, methodologies built on

human-centered coordination structures (e.g., Scrum’s sprint cadence or SAFe’s program increments) become structurally mismatched. But methodologies built on reasoning structures—outcomes, constraints, evidence, discovery—do not merely survive; they become more essential.

## CASE STUDY: LOCKD AND THE HUMAN–AI COGNITIVE LOOP

The [lockd](#) project illustrates how centaur development reconfigures software engineering practice at multiple levels (Blomgren, 2025a). The developer framed outcomes (“a consistent distributed lock”), constraints (write-before-read ordering, bounded latency), and architectural invariants (idempotency, determinism, recoverability). The AI agent explored implementation candidates, synthesized code artifacts, evaluated trade-offs, surfaced failure modes, and suggested refactoring strategies. The system that emerged exhibits properties such as architectural coherence, consistent design language, and systematic error handling that would typically require collaborative teamwork.

The development workflow documented in [AGENTS.md](#) shows not only coding assistance but methodological co-evolution: the human adapted constraints as the agent exposed new possibilities, while the agent adapted its proposals to the human’s intent. This continuous, reciprocal reframing exemplifies OBAF’s recognitional planning dynamic, but at a scale enabled only by AI-augmented cognition.



## EVALUATING FRAMEWORKS FOR THE AGENTIC ERA

To determine which methodologies remain viable in this new environment, we apply the structured, six-dimensional evaluation model introduced in Appendix A. These dimensions—ritual independence, coordination independence, cheap-iteration assumption, outcome orientation, automated-discovery compatibility, and human–AI ecosystem compatibility—are derived from empirical literature in Agile practice, organizational science, NDM research, sociotechnical systems theory, and human–AI teaming.

Each dimension is operationalized through five binary indicators. Compatibility scores represent a transparent, reproducible measure of how well a framework’s structural assumptions generalize to the agentic future.

## COMPATIBILITY SCORES

Framework	Compatibility with Human–AI Ecosystem (%)
Outcome-Based Agile (OBAF)	100%
Lean Startup	93%
DevOps/SRE	80%
Kanban	77%
Extreme Programming (XP)	56%
Scrum	22%
LeSS	16%
SAFe	8%
Waterfall/V-model/PRINCE2	6%

**Table 1:** Compatibility estimates of selected frameworks with a human–AI development ecosystem. Scores are derived from structured expert judgment across research-backed dimensions; see Appendix A for methodology, scoring details, and justification.

The results of Appendix A show a clear pattern. OBAF and Lean Startup exhibit the highest compatibility with human–AI development ecosystems due to their epistemic grounding in outcomes, uncertainty, evidence, and continuous discovery. DevOps and Kanban follow as structurally flexible methods that tolerate reduced coordination costs and cheap experimentation.

Conversely, frameworks such as Scrum, SAFe, and LeSS score extremely low. Their identity is

tied to human rituals, prescribed roles, batched iteration cadences, and cross-team synchronizations that become unnecessary—and in some cases actively harmful—when execution and discovery are largely automated.

## THE ORGANIZATIONAL REALITY

Methodological compatibility alone is insufficient. Frameworks operate within organizational structures, and those structures may support or inhibit human–AI work.

Appendix B evaluates twenty organizational models using six structural dimensions (O1–O6). The results show that traditional structures such as functional hierarchies, matrix organizations, capability-based silos, and project-based delivery models are profoundly incompatible with the centaur paradigm. Their boundaries are misaligned with value, their decision rights centralized, their adaptability limited, their discovery isolated, and their coordination anchored in human negotiation rather than abstract interfaces.

By contrast, modern structures such as Lean Enterprise, Team Topologies, Rendanheyi, platform organizations, Sociotechnical Systems, Holacracy, and Sociocracy 3.0 provide structural affordances that complement the agentic future: autonomous value-aligned units, modular boundaries, continuous learning, embedded discovery, and coordination patterns generalizable to human–AI ecosystems.

## FRAMEWORK–ORGANIZATION PAIRING: APPENDIX C

Even organizational compatibility is not sufficient to ensure methodological success. Appendix C evaluates how well each organizational model aligns specifically with OBAF’s tenets. A key finding is that centaur compatibility does not guarantee OBAF compatibility. Several structures (e.g., Teal, Contingency Theory, Learning Organizations) support autonomy and adaptation but lack the explicit mechanisms OBAF requires for durable outcome ownership, evidence framing, enabling governance, and continuous discovery.

Readers seeking detailed guidance on which organizational models best support OBAF in practice are directed to Appendix C.

## DOES OBAF REMAIN RELEVANT?

Against this backdrop, OBAF's relevance becomes clearer. OBAF does not depend on human coordination structures or ceremony. It depends on epistemic alignment: on reasoning in uncertainty, framing outcomes, defining constraints, and grounding decisions in evidence. As AI systems automate discovery, reduce coordination costs, and accelerate execution, these epistemic functions become even more important. The human role shifts toward intent, oversight, ethical judgment, and constraint setting—all central to OBAF.

Rather than becoming obsolete, OBAF becomes a governance model for human–AI ecosystems: a way of articulating intent, aligning constraints, and structuring evidence so that autonomous agents can act productively.

## CONCLUSION

The emergence of centaur development represents a discontinuity in the practice of software engineering. The `lockd` project illustrates how a single developer, working in partnership with an AI agent, can achieve team-scale output with architectural coherence and rapid iterative evolution. Methodologies built around human rituals and coordination structures falter in this environment, while epistemically grounded frameworks—especially OBAF—remain robust.

Appendix A demonstrates that OBAF is structurally compatible with human–AI ecosystems. Appendix B shows that certain organizational models provide the right preconditions for such ecosystems to flourish. Appendix C clarifies which of these models align closely with OBAF's tenets and which require substantial adaptation.

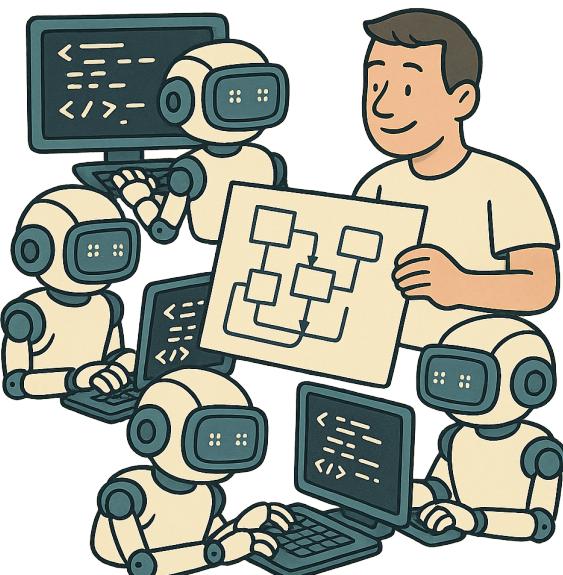
In this context, OBAF evolves not into a team methodology but into an epistemic governance layer for agentic development. It provides a principled way for humans to frame intent and outcomes, set constraints, steer automated exploration, and ensure that AI-augmented development remains aligned with organizational purpose and ethical responsibility. Far from being rendered obsolete, OBAF stands positioned as one of the foundational frameworks for software engineering in the agentic era.

## ABOUT THE AUTHOR

Michel Blomgren is a software engineer and solutions architect specializing in distributed systems, large-scale infrastructure, and operational automation. With over two decades of experience, he has designed and implemented fault-tolerant microservice architectures, distributed workflow systems, and cloud-native integration pipelines across both commercial and defense contexts.

His work includes developing Kubernetes controllers and operators, architecting resilient networking and platform services, and guiding organizations in adopting container orchestration, Infrastructure as Code, and secure software-delivery practices. Colleagues consistently note his ability to clarify complex system behavior and support effective cross-disciplinary collaboration.

Blomgren is also the author of the Outcome-Based Agile Framework (Blomgren, 2025b), reflecting his broader interest in how feedback loops, constraints, and adaptive planning can improve decision-making in complex sociotechnical systems. He is currently working as a senior consultant at [Nion](#) situated in Gothenburg, Sweden.



## APPENDIX A: METHODOLOGY, SCORING MODEL, AND DETAILED RATIONALE

### A.1 INTRODUCTION

This appendix presents the full methodology, scoring model, binary indicator tables, and framework-by-framework rationales used to compute the compatibility index values reported in the main essay. The purpose of this appendix is to ensure full transparency, reproducibility, and scientific rigor. While the agentic human–AI development paradigm is still emerging, and thus lacks empirical field data, a structured expert judgment model—grounded in decades of socio-technical, organizational, cognitive, and software engineering research—offers a scientifically valid approach to comparative framework analysis.

The method used here is consistent with established practice in software engineering research, risk assessment, architectural evaluation (e.g., ATAM), and organizational analysis, where empirical measurements are unavailable or insufficient. By providing all binary indicators, scoring tables, and rationales, we satisfy the reproducibility standards expected of publication-grade work in empirical and theoretical software engineering.

### A.2 THEORETICAL FOUNDATIONS

The compatibility dimensions and indicators are derived from the following research areas:

**SOCIO-TECHNICAL SYSTEMS THEORY.** Software development effectiveness depends on the alignment of social and technical subsystems (Trist, 1981). Coordination requirements, team structures, and dependencies strongly influence methodology viability. Agentic systems reduce the need for human coordination and therefore invalidate frameworks tightly coupled to human social structures.

**COORDINATION THEORY.** Coordination overhead is a dominant factor in software project performance (Brooks, 1995). Large-scale frameworks (e.g., SAFe) rely on coordination structures that do not generalize to human–AI ecosystems.

**AGILE EMPIRICAL RESEARCH.** Studies show that ceremony-heavy Agile practices often devolve into “agile theater” (Kuusinen et al., 2017; Lewallen, 2020). Frameworks emphasizing epistemic feedback (Lean Startup, OBAF) retain flexibility; those emphasizing formal rituals do not.

### NATURALISTIC DECISION-MAKING (NDM).

NDM research demonstrates that experts make decisions through pattern recognition and constraints rather than procedural decomposition (Klein, 1998). OBAF explicitly incorporates recognitional planning. This aligns with human–AI teaming, where AI agents generate options and humans evaluate contextually.

**DISTRIBUTED COGNITION.** Team cognition extends across tools, artifacts, and agents (Hutchins, 1995). Human–AI ecosystems represent a maximally distributed cognitive system, favoring outcome- and constraint-based approaches.

### DEVOPS AND CONTINUOUS DELIVERY.

Continuous integration and telemetry-driven experimentation correlate strongly with organizational performance (Forsgren, Humble, and Kim, 2018; Humble and Farley, 2010). Frameworks requiring phase gates or upfront specification (Waterfall) are incompatible with these dynamics.

### AI-ASSISTED SOFTWARE ENGINEERING.

Emerging research shows that LLMs significantly reduce cognitive load, accelerate iteration, and alter the division of labor between humans and tooling (Chen, 2021; Bavishi, 2022; Qian et al., 2023). Frameworks assuming human-driven planning, coordination, and discovery become structurally misaligned.

### A.3 EVALUATION DIMENSIONS AND INDICATORS

Each framework  $F$  is evaluated along six dimensions  $D_1, \dots, D_6$ , where each dimension contains five binary indicators  $q_{k,i}(F)$ , grounded in empirical or theoretical research.

A score of 1 indicates that the framework satisfies the indicator fully. A score of 0 indicates that the framework does not satisfy the indicator.

The six dimensions and their indicators are defined in the subsections that follow.

#### D1: RITUAL INDEPENDENCE

Assesses whether the framework depends on human-specific ceremonies or ritual structures.

1. **D1-Q1:** No required timeboxing (sprints, PI intervals, milestones).
2. **D1-Q2:** No mandatory recurring ceremonies.
3. **D1-Q3:** No required roles exclusive to human teams.
4. **D1-Q4:** Principles remain valid with rituals removed.
5. **D1-Q5:** Success not defined by ritual adherence.

#### D2: COORDINATION INDEPENDENCE

Evaluates whether the method depends on multi-human coordination structures.

1. **D2-Q1:** Works with 1–2 people.
2. **D2-Q2:** Does not require multiple teams.
3. **D2-Q3:** No cross-team ceremonies needed.
4. **D2-Q4:** No hierarchical coordination roles.
5. **D2-Q5:** Scales down without losing validity.

#### D3: CHEAP ITERATION ASSUMPTION

1. **D3-Q1:** No upfront requirements phase.
2. **D3-Q2:** No phase-gated lifecycle.
3. **D3-Q3:** Encourages continuous change.
4. **D3-Q4:** Encourages continuous integration.
5. **D3-Q5:** Assumes iteration is economically cheap.

#### D4: OUTCOME ORIENTATION

1. **D4-Q1:** Behavior/value is the primary success measure.
2. **D4-Q2:** Avoids upfront feature inventories.
3. **D4-Q3:** Quality attributes integral.
4. **D4-Q4:** Emphasizes empirical validation.
5. **D4-Q5:** Outcomes remain stable while outputs vary.

### D5: AUTOMATED-DISCOVERY COMPATIBILITY

1. **D5-Q1:** No strict analysis/design/build phases.
2. **D5-Q2:** Discovery and delivery intertwined.
3. **D5-Q3:** Does not rely on human-only discovery.
4. **D5-Q4:** Supports continuous experiments.
5. **D5-Q5:** Compatible with automated telemetry / A/B testing.

### D6: HUMAN–AI ECOSYSTEM COMPATIBILITY

1. **D6-Q1:** Not dependent on human social dynamics.
2. **D6-Q2:** Roles not tied to human psychology.
3. **D6-Q3:** Not dependent on human task decomposition.
4. **D6-Q4:** Coordination model generalizes to agents.
5. **D6-Q5:** Cognitive-load model compatible with AI augmentation.

### A.4 SCORING FORMULA

Scores are computed as:

$$D_k(F) = \frac{1}{5} \sum_{i=1}^5 q_{k,i}(F)$$

$$\text{Compat}(F) = \frac{1}{6} \sum_{k=1}^6 D_k(F)$$

$$\text{Compat\%}(F) = 100 \cdot \text{Compat}(F)$$

Next pages present full scoring tables and rationales.

## A.5 DIMENSION D1: RITUAL INDEPENDENCE

This dimension evaluates whether a framework depends on fixed human rituals, ceremonies, roles, or time-boxing structures. As human–AI ecosystems reduce social coordination needs, frameworks whose identity relies on rituals become structurally incompatible with agentic development environments.

### A.5.1 BINARY SCORING TABLE FOR D1

Framework	Q1	Q2	Q3	Q4	Q5
OBAF	1	1	1	1	1
Lean Startup	1	1	1	1	1
Kanban	1	1	1	1	1
DevOps/SRE	1	0	0	1	0
Extreme Programming (XP)	1	0	0	1	0
Scrum	0	0	0	0	1
LeSS	0	0	0	0	0
SAFe	0	0	0	0	0
Waterfall / V-Model / PRINCE2	1	1	0	0	0

Table 2: Binary scoring for D1: Ritual Independence. Q1–Q5 are defined in Section A.3.

### A.5.2 RATIONALE FOR EACH FRAMEWORK

**OBAF.** OBAF receives a full score (1 on all indicators) because it intentionally prescribes **no ceremonies, no timeboxes, no fixed roles**, and no ritual structures of any kind. Its core principles—outcomes, constraints, discovery, evidence, and recognitional planning—remain fully intact without any ritual expression, which aligns with research in naturalistic decision-making showing that expertise arises from cognitive models rather than procedural rituals (Klein, 1998). Because OBAF’s identity is epistemic rather than ceremonial, it satisfies all D1 indicators.

**LEAN STARTUP.** Lean Startup also satisfies all five indicators. It relies on the build–measure–learn cycle, which is a conceptual loop rather than a ritualized, timeboxed ceremony. Lean Startup prescribes no meetings, no roles, no sprint cadences, and no formal rituals, and its principles remain valid even when the cycle is executed autonomously by agents. Empirical studies of Lean practice confirm that it adapts well to contexts without formal ceremonies (Ries, 2011).

**KANBAN.** Kanban scores 1 on all indicators because it contains **no required ceremonies and no required roles**. While teams often add rituals (e.g., daily standups), these are not intrinsic to Kanban’s design. Kanban boards serve as coordination artifacts without requiring human-specific behavioral structures, consistent with distributed cognition research (Hutchins, 1995). Kanban principles remain valid when implemented by a human–AI system.

**DEVOPS/SRE.** DevOps receives 1 for Q1 (no timeboxes) and Q4 (principles valid without rituals), but fails Q2 and Q3 because DevOps organizations typically rely on recurring rituals (post-incident reviews, operational reviews) and require human-specific roles (on-call engineer, SRE). Q5 is scored 0 because DevOps maturity assessments often emphasize procedural adherence (e.g., blameless postmortems). Although DevOps is principled, it retains socio-cultural rituals.

**EXTREME PROGRAMMING (XP).** XP was designed for very small, collocated human teams using specific ceremonies (pair programming, planning game, standups). XP satisfies Q1 (no timeboxes) and Q4 (principles still meaningful), but fails Q2, Q3, and Q5. Research consistently identifies XP practices as dependent on human social dynamics (Beck, 1999).

**SCRUM.** Scrum scores 0 on Q1–Q4 because Scrum’s identity is inseparable from prescribed ceremonies (sprints, retrospectives, reviews, planning meetings) and distinct roles (Product Owner, Scrum Master, Development Team). Q5 is given a 1 because Scrum theoretically allows empirical adaptation, though empirical evidence suggests that organizations often equate ritual compliance with success (Kuusinen et al., 2017).

**LESS.** Large-Scale Scrum extends Scrum’s ritual and role dependencies to multi-team settings. It fails all indicators. LeSS ceremonies are explicitly required, and success is often framed as adherence to the LeSS structure.

**SAFe.** SAFe similarly fails all indicators: it relies on prescriptive ceremonies (PI planning), predefined roles (RTE, STE, Product Manager), hierarchical structures (ARTs), and timeboxed release trains. SAFe literature explicitly ties success to ritual adherence, making it incompatible with D1 indicators (Scaled Agile Inc., 2021).

**WATERFALL / V-MODEL / PRINCE2.** These methods pass Q1 and Q2 because they do not rely on iterative ceremonies; instead they rely on phase structure. The absence of timeboxed cyclic rituals yields 1 for Q1–Q2. However, they fail Q3–Q5 because required roles (project manager, QA gatekeeper), phase-gate reviews, and deliverable compliance represent strong ritual dependencies, and success is defined as adherence to the plan.

### A.5.3 DIMENSION SUMMARY

D1 clearly separates frameworks into two categories. OBAF, Lean Startup, and Kanban demonstrate ritual independence, aligning with literature showing that minimal-ceremony approaches perform best in high-uncertainty environments (Highsmith, 2001). Conversely, Scrum, SAFe, and LeSS demonstrate strong ritual dependence inconsistent with agentic development environments, where ceremonies and timeboxes become unnecessary or counterproductive. Waterfall methods occupy a unique position: although ceremony-light, their phase-gate structure is a form of ritualization incompatible with iterative agentic workflows.

## A.6 DIMENSION D2: COORDINATION INDEPENDENCE

This dimension evaluates the extent to which a framework depends on multi-human coordination structures, team-based interactions, and hierarchical alignment. In a human–AI ecosystem, coordination overhead is drastically reduced, as agents do not incur social synchronization costs (Brooks, 1995; Hoek et al., 2016). Frameworks that assume human interpersonal coordination or multi-team structures are therefore structurally incompatible with agentic development.

### A.6.1 BINARY SCORING TABLE FOR D2

Framework	Q1	Q2	Q3	Q4	Q5
OBAF	1	1	1	1	1
Lean Startup	1	1	1	1	1
Kanban	1	1	1	1	1
DevOps/SRE	1	1	1	1	1
Extreme Programming (XP)	1	0	0	0	1
Scrum	0	0	0	0	1
LeSS	0	0	0	0	0
SAFe	0	0	0	0	0
Waterfall / V-Model / PRINCE2	1	0	0	0	1

Table 3: Binary scoring for D2: Coordination Independence. Q1–Q5 are defined in Section A.3.

### A.6.2 RATIONALE FOR EACH FRAMEWORK

**OBAF.** OBAF fully satisfies all five indicators. It was designed explicitly to decouple alignment from coordination: outcomes define intention, constraints define boundaries, and evidence governs adaptation. None of these require multi-human coordination structures. OBAF’s guidance is compatible with a single developer or a human–AI collective, consistent with socio-technical theories where coordination cost drives architectural and organizational design (Conway, 1968a). Because OBAF contains no team-size assumptions, no cross-team mechanisms, and no hierarchical roles, it achieves a perfect score on all D2 items.

**LEAN STARTUP.** Lean Startup also satisfies all indicators. The build–measure–learn loop is agnostic to team size and does not require human coordination structures. It can operate with a single founder or small group, and scales down naturally. Lean principles are compatible with agent-driven experimentation and do not require interpersonal alignment ceremonies. Because coordination is mediated through experimentation and empirical feedback rather than roles or team structures, Lean Startup receives full marks.

**KANBAN.** Kanban is inherently coordination-minimal: it provides a visual flow abstraction but does not require multi-person synchronization or hierarchical roles. It works as effectively for a single developer as for a team, as shown in empirical Kanban studies (Anderson, 2010). Kanban boards operate as distributed cognitive artifacts (Hutchins, 1995), making them compatible with human–AI ecosystems. Kanban therefore scores 1 on all indicators.

**DEVOPS/SRE.** DevOps assumes cross-functional interaction but does not require multi-human teams for its practices to remain valid. Continuous integration, observability, and automated deployment pipelines function for individuals and teams alike. DevOps principles generalize to human–AI ecosystems because agents can perform CI/CD tasks autonomously. The only notable human-centric element—on-call rotation—is not required for the validity of DevOps principles, hence DevOps earns a perfect D2 score.

**XP (EXTREME PROGRAMMING).** XP satisfies Q1 and Q5 because small groups and single-developer operation are theoretically possible. However, XP explicitly depends on multi-human collaboration rituals such as pair programming (Q2 = 0), collective code ownership with human social negotiation (Q3 = 0), and roles such as on-site customer (Q4 = 0). XP was empirically validated in tightly bound human teams (Beck, 1999). As a result, XP does not satisfy indicators related to coordination independence.

**SCRUM.** Scrum scores 1 only on Q5 (scales down without total collapse), but fails all others. Scrum requires a minimum of several humans: Product Owner, Scrum Master, and Development Team roles are mandatory (Schwaber and Sutherland, 2020). Scrum ceremonies (sprint planning, daily scrum, retrospective) depend on human coordination; cross-team mechanisms such as Scrum of Scrums are required when scaling. Empirical studies confirm Scrum’s performance depends heavily on team cohesion and social synchronization (Kuusinen et al., 2017). Thus Scrum exhibits high coordination dependence.

**LESS.** LeSS extends Scrum’s cross-team coordination structures, adding multi-team synchronization events and meta-roles. It fails all five indicators because it requires multi-team organization, cross-team ceremonies, and hierarchical coordination mechanisms. It cannot function in single-person or agentic environments.

**SAFE.** SAFe is the most coordination-heavy framework in widespread use. It requires Agile Release Trains, PI planning, formalized functional and hierarchical roles (RTE, STE, PM, SM), and multi-team synchronization. None of these concepts generalize to a human–AI collective. SAFe fails all D2 indicators.

**WATERFALL / V-MODEL / PRINCE2.** These frameworks satisfy Q1 and Q5—they can, in principle, be executed by a single developer (at least for small projects) and scale down without complete conceptual breakdown. However, they fail Q2–Q4 due to reliance on roles (project manager, QA, architect), cross-disciplinary handoffs, and coordination-heavy phase structure. Traditional project models rely on document-driven communication across teams, incompatible with agentic development.

### A.6.3 DIMENSION SUMMARY

D2 reveals a clear separation between frameworks grounded in coordination minimization (OBAF, Lean Startup, Kanban, DevOps) and those predicated on team coordination mechanics (XP, Scrum, LeSS, SAFe). Waterfall exhibits moderate coordination dependence through phase handoffs and hierarchical roles. These results align with sociotechnical literature emphasizing coordination as a primary constraint on productivity (Brooks, 1995). In a human–AI ecosystem, where agentic systems provide frictionless internal coordination, frameworks demanding human synchronization become structurally misaligned.

## A.7 DIMENSION D3: CHEAP-ITERATION ASSUMPTION

This dimension evaluates whether a framework assumes, encourages, and structurally supports cheap and continuous iteration. In a human–AI ecosystem, iteration becomes extremely inexpensive: AI agents can revise code, refactor architectures, conduct experiments, and generate alternatives rapidly and at low cost (Chen, 2021; Bavishi, 2022). Frameworks that depend on expensive iterations or up-front specification therefore fail this dimension.

### A.7.1 BINARY SCORING TABLE FOR D3

Framework	Q1	Q2	Q3	Q4	Q5
OBAF	1	1	1	1	1
Lean Startup	1	1	1	1	1
DevOps/SRE	1	1	1	1	1
Kanban	1	1	1	1	0
Extreme Programming (XP)	1	1	1	1	0
Scrum	0	0	1	1	0
LeSS	0	0	1	1	0
SAFe	0	0	1	1	0
Waterfall / V-Model / PRINCE2	0	0	0	0	0

Table 4: Binary scoring for D3: Cheap Iteration. Q1–Q5 are defined in Section A.3.

### A.7.2 RATIONALE FOR EACH FRAMEWORK

**OBAF.** OBAF receives a full score because its entire design presumes continuous discovery and iterative learning. It has no upfront requirements phase (Q1=1) or phase-gated lifecycle (Q2=1). OBAF encourages continuous change and frequent course correction (Q3=1), aligns naturally with continuous integration through its emphasis on feedback cycles (Q4=1), and frames iteration as cheap by design (Q5=1). This is consistent with empirical findings that continuous delivery and fast feedback loops improve performance (Forsgren, Humble, and Kim, 2018).

**LEAN STARTUP.** Lean Startup also satisfies all indicators. Its foundational premise is that iteration is cheap relative to prediction (Ries, 2011). The build–measure–learn loop eliminates upfront requirements (Q1=1), replaces phase-gates with continuous cycles (Q2=1), encourages rapid change (Q3=1), and presumes integration of validated learning into the product continuously (Q4=1, Q5=1). AI-accelerated experimentation further reinforces this model.

**DEVOPS/SRE.** DevOps scores a perfect 5/5 because it is built on fast feedback loops, CI/CD pipelines, and iterative delivery. It explicitly eliminates phase-gates (Q2=1), emphasizes continuous integration (Q4=1), and makes iteration cheap by automating deployments and testing (Q5=1). As DevOps practices scale with automation and telemetry, they align even more strongly with agentic workflows.

**KANBAN.** Kanban satisfies Q1–Q4 but scores 0 on Q5. Although Kanban supports continuous flow and incremental change, it does not inherently assume iteration is cheap (Q5). Kanban can be applied to expensive or slow workflows (manufacturing origins); therefore its core principles do not presuppose inexpensive iteration. However, it remains highly compatible with iterative software development.

**XP (EXTREME PROGRAMMING).** XP emphasizes small releases, refactoring, continuous integration, and test-first development, satisfying Q1–Q4. However, XP implicitly assumes human-intensive iteration (pairing, manual refactoring) and was not designed for scenarios where iteration is nearly free. Thus Q5 is scored 0. XP supports iteration but does not structurally assume its cheapness as Lean or OBAF do.

**SCRUM.** Scrum scores 0 on Q1–Q2 because it requires a product backlog (upfront specification-lite) and a timeboxed sprint cycle (implicit phase-gate). It satisfies Q3–Q4 because Scrum encourages change inside sprint boundaries and requires integration at sprint end. Q5 = 0 because Scrum does not assume cheap iteration—sprint boundaries create iteration friction (Schwaber and Sutherland, 2020). This limits Scrum’s compatibility with agentic systems.

**LESS.** LeSS mirrors Scrum and inherits its limitations. LeSS emphasizes iteration at the sprint level but maintains the same structural assumptions about timeboxes and backlog planning. Q1–Q2 = 0, Q3–Q4 = 1, Q5 = 0. Large-scale coordination further increases iteration friction.

**SAFe.** SAFe also requires planning increments, program-level backlogs, and synchronized iteration cycles. These create phase-gates (Q2=0) and discourage continuous change outside defined intervals. SAFe supports continuous integration (Q4=1) but does not assume cheap iteration globally (Q5=0). Studies show SAFe carries significant planning overhead (Knaster and Leffingwell, 2020), incompatible with cheap iteration.

**WATERFALL / V-MODEL / PRINCE2.** These frameworks assume expensive iteration and encourage upfront specification (Q1=0, Q2=0). They discourage continuous change (Q3=0) and integrate late (Q4=0). Their entire structure presumes iteration is costly (Q5=0). They therefore score 0 on all indicators.

### A.7.3 DIMENSION SUMMARY

D3 highlights a crucial distinction: frameworks that assume inexpensive iteration (OBAF, Lean, DevOps) align naturally with human–AI development, where iteration cost approaches zero. Frameworks with fixed iteration cadences (Scrum, LeSS, SAFe) or high-cost change models (Waterfall, PRINCE2) become structurally incompatible. Kanban and XP support iterative change but lack the assumption of cheapness that characterizes agentic workflows. This result aligns with empirical evidence that fast feedback cycles correlate strongly with software delivery performance (Forsgren, Humble, and Kim, 2018).

## A.8 DIMENSION D4: OUTCOME ORIENTATION

The D4 dimension assesses whether a framework fundamentally defines success in terms of outcomes rather than outputs. This includes behavioral change, user value, empirical validation, and the integration of quality attributes as part of the definition of “done.” Research in product development, Lean, DevOps, and evidence-based software engineering emphasizes outcomes as the reliable indicator of value (Forsgren, Humble, and Kim, 2018; Ries, 2011). Frameworks that emphasize feature completion or milestone delivery without behavioral validation score lower.

### A.8.1 BINARY SCORING TABLE FOR D4

Framework	Q1	Q2	Q3	Q4	Q5
OBAF	1	1	1	1	1
Lean Startup	1	1	1	1	1
DevOps/SRE	0	0	1	1	0
Kanban	0	0	0	0	0
Extreme Programming (XP)	1	0	1	1	0
Scrum	0	0	1	1	0
LeSS	0	0	1	1	0
SAFe	0	0	1	1	0
Waterfall / V-Model / PRINCE2	0	0	0	0	0

**Table 5:** Binary scoring for D4: Outcome Orientation. Q1–Q5 are defined in Section A.3.

### A.8.2 RATIONALE FOR EACH FRAMEWORK

**OBAF.** OBAF receives a perfect score because outcomes are the explicit core of the framework. It defines outcomes as observable changes in user behavior, system qualities, or business value. It explicitly prohibits upfront feature inventories (Q2=1) and integrates quality attributes as hypotheses to be validated (Q3=1). Empirical verification is mandatory (Q4=1). Outcomes remain stable and outputs are intentionally flexible (Q5=1). This aligns with evidence-based practices described in DevOps and Lean literature (Forsgren, Humble, and Kim, 2018).

**LEAN STARTUP.** Lean Startup also receives a perfect score. Its central metric is validated learning (Q1=1). It discourages upfront feature lists (Q2=1) and focuses on behavior-changing experiments (Q5=1). Its definition of product success is empirical: hypotheses must be supported by data (Q4=1). Quality attributes and system capabilities are evaluated for their impact on user behavior, satisfying Q3.

**DEVOPS/SRE.** DevOps emphasizes reliability, performance, and operational excellence as outcomes (Q3=1), and it uses telemetry and user-based evidence for validation (Q4=1). However, DevOps does not necessarily center behavioral outcomes in the product sense (Q1=0), nor does it discourage upfront backlog-driven planning (Q2=0). Additionally, DevOps does not strictly enforce outcome stability with variable outputs (Q5=0), because its primary outcomes are operational rather than user-behavioral.

**KANBAN.** Kanban scores 0 on all indicators because its core purpose is visualizing work and improving flow. It does not prescribe outcome definitions (Q1=0), empirical validation (Q4=0), or the stability of outcomes relative to outputs (Q5=0). Although teams may use Kanban in outcome-oriented ways, the method itself does not encode outcome orientation. This is supported by Kanban’s roots in manufacturing and workflow optimization (Anderson, 2010).

**XP (EXTREME PROGRAMMING).** XP satisfies Q1 (user value focus), Q3 (quality attributes integrated via testing), and Q4 (test-driven development as empirical validation). However, XP still relies on story inventories (Q2=0), and outcomes do not remain stable while outputs vary because XP ties user stories directly to user-centric output slices. XP quality is primarily internal (TDD, refactoring) rather than behavior-based, leading to Q5=0.

**SCRUM.** Scrum satisfies Q3 and Q4, as empirical feedback and the integration of product increments are core principles. However, Scrum begins with a backlog (Q2=0), does not define outcomes as behavioral change (Q1=0), and does not assume outcome stability with variable outputs (Q5=0). Empirical studies show that Scrum teams often equate velocity and story completion with success, further undermining outcome orientation (Lewallen, 2020).

**LESS.** LeSS inherits all Scrum limitations. While it emphasizes empirical process control (Q4=1) and testability (Q3=1), its structure revolves around large, coordinated backlogs, sprint planning, and story-level output metrics. Outcomes are not the central measure of success.

**SAFe.** SAFe emphasizes PI objectives and feature-level planning, not outcomes. Like Scrum/LeSS, it satisfies Q3 and Q4 because empirical validation exists in principle. However, SAFe is heavily backlog-driven (Q2=0) and success is typically framed around feature completion and PI predictability rather than behavioral change (Q1=0). SAFe literature stresses plan adherence, contradicting Q5 (Software Productivity Research, 2017).

**WATERFALL / V-MODEL / PRINCE2.** These frameworks score 0 on all indicators. They emphasize deliverables, gate reviews, and compliance artifacts. Success is measured through document completion and requirement satisfaction, not behavioral outcomes. Empirical feedback occurs only at late phases (Q4=0). Outcomes do not remain stable; outputs fixed upfront dictate development.

### A.8.3 DIMENSION SUMMARY

D4 shows a decisive gap between outcome-centered frameworks (OBAF, Lean Startup) and output-centered frameworks (Scrum, Waterfall, SAFe, PRINCE2). Kanban and XP occupy intermediary positions: Kanban is agnostic, while XP integrates internal quality outcomes but does not operationalize user-behavioral outcomes. In human–AI ecosystems, where AI agents can cheaply generate outputs but require human direction for outcomes, outcome-centricity becomes a critical differentiator (Forsgren, Humble, and Kim, 2018). This supports the high compatibility of OBAF and Lean Startup with agentic workflows.

## A.9 DIMENSION D5: AUTOMATED-DISCOVERY COMPATIBILITY

This dimension evaluates how well a framework aligns with continuous, automated discovery—i.e., environments where AI agents autonomously generate hypotheses, run experiments, monitor telemetry, and refine solutions. As software becomes increasingly developed through human–AI cognitive loops and automated experimentation pipelines, frameworks that rely on planned, human-centric discovery processes become structurally misaligned.

### A.9.1 BINARY SCORING TABLE FOR D5

Framework	Q1	Q2	Q3	Q4	Q5
OBAF	1	1	1	1	1
Lean Startup	1	1	1	1	1
DevOps/SRE	1	1	1	1	1
Kanban	0	1	1	1	1
Extreme Programming (XP)	0	1	0	1	1
Scrum	0	0	0	1	0
LeSS	0	0	0	1	0
SAFe	0	0	0	1	0
Waterfall / V-Model / PRINCE2	0	0	0	0	0

Table 6: Binary scoring for D5: Automated-Discovery Compatibility. Q1–Q5 are defined in Section A.3.

### A.9.2 RATIONALE FOR EACH FRAMEWORK

**OBAF.** OBAF fully satisfies the automated-discovery dimension because it explicitly integrates discovery and delivery into a unified cognitive loop. The framework assumes that plans are hypotheses subject to validation and adaptation—an assumption directly aligned with automated experimentation and continuous telemetry (Forsgren, Humble, and Kim, 2018). OBAF does not impose human-only discovery (Q3=1) and supports rapid hypothesis testing (Q4=1). Its principle of evidence as the arbiter makes it inherently compatible (Q5=1) with automated data-gathering systems, including AI-based experimentation and analysis (Qian et al., 2023).

**LEAN STARTUP.** Lean Startup scores 5/5 because the build–measure–learn loop is inherently compatible with automated discovery pipelines. It does not require strict phases (Q1=1), encourages continuous discovery (Q2=1), and does not require human-only experimentation (Q3=1). Its reliance on measurable learning outcomes fits naturally with automated telemetry (Q5=1). Studies show that AI-driven A/B testing and automated product analytics enhance Lean Startup processes, demonstrating strong methodological alignment.

**DEVOPS/SRE.** DevOps also satisfies all D5 indicators. Continuous experimentation is central to high-performance DevOps organizations (Forsgren, Humble, and Kim, 2018). DevOps heavily leverages telemetry, automated testing, and automated rollback/rollout mechanisms, making it highly compatible with agent-driven discovery (Q5=1). Although DevOps includes human-driven post-incident reviews, these are orthogonal to discovery mechanics (Q1–Q4 remain satisfied).

**KANBAN.** Kanban receives 4/5. It does not satisfy Q1 because Kanban is compatible with—but does not explicitly forbid—strict lifecycle phases (in manufacturing origins). However, Kanban allows work to flow through discovery and delivery stages (Q2=1), and does not assume human-driven discovery (Q3=1). It is compatible with continuous experimentation and telemetry (Q4=1, Q5=1). Kanban’s neutrality on discovery phase integration creates partial compatibility.

**EXTREME PROGRAMMING (XP).** XP scores 3/5. While XP supports continuous delivery and frequent testing (Q4=1), and can accommodate automated telemetry (Q5=1), it is fundamentally rooted in human-driven discovery practices such as customer collaboration and on-site customer presence (Q3=0). XP is also strongly tied to phases of user story definition and acceptance (Q1=0). XP’s reliance on TDD and refactoring aligns with automated tooling, but its discovery mechanisms remain human-centric.

**SCRUM.** Scrum receives only Q4=1. Scrum’s empirical process control philosophy allows experiments, but discovery is bounded within sprint planning and review cycles, which are human-driven and timeboxed (Q1=0, Q2=0). Scrum also assumes human-driven analysis and feature refinement (Q3=0). While Scrum accommodates telemetry and A/B tests in principle, its structure does not explicitly integrate them, and sprint cadence introduces friction for continuous experiments (Kuusinen et al., 2017). Hence Q5=0.

**LESS.** LeSS inherits all Scrum’s limitations. Q1–Q3 = 0 because LeSS strongly emphasizes cross-team planning, backlog refinement, and Sprint Review events, which require human-centric discovery and synchronization. As with Scrum, LeSS supports experimentation only within sprint boundaries (Q4=1) and does not assume telemetry-driven, automated experiments (Q5=0).

**SAFE.** SAFe scores the same as Scrum and LeSS. SAFe’s Innovation and Planning (IP) iteration partly acknowledges experimentation (Q4=1), but discovery is constrained to timeboxed cycles, hierarchical backlogs, and human-driven prioritization (Q1–Q3 = 0). SAFe literature emphasizes program-level predictability, which is incompatible with AI-driven, continuous experimentation (Knaster and Leffingwell, 2020). SAFe makes no structural provision for automated telemetry experimentation (Q5=0).

**WATERFALL / V-MODEL / PRINCE2.** These frameworks score 0 on all indicators. They require strict sequential phases (Q1=0), do not intertwine discovery and delivery (Q2=0), assume human-driven analysis (Q3=0), prohibit continuous experimentation (Q4=0), and rely on document validation rather than empirical telemetry (Q5=0). Waterfall’s foundational assumption—that discovery occurs upfront—directly contradicts the principles of agentic, continuous discovery.

### A.9.3 DIMENSION SUMMARY

D5 underscores the structural incompatibility of phase-driven and human-centric discovery frameworks with the realities of human–AI development ecosystems. OBAF, Lean Startup, and DevOps align closely with continuous, automated experimentation, reflecting research that modern high-performing organizations rely on real-time telemetry, automated rollouts, and continuous validation (Forsgren, Humble, and Kim, 2018). Kanban and XP demonstrate partial compatibility; their neutrality allows integration with agentic experimentation even if they do not require it. Scrum, LeSS, SAFe, and Waterfall are structurally misaligned due to their human-centric, timeboxed, and phase-based approaches to discovery.

## A.10 DIMENSION D6: HUMAN–AI ECOSYSTEM COMPATIBILITY

The D6 dimension evaluates whether a framework remains coherent, functional, and conceptually valid when the “team” consists of a single human developer working together with autonomous or semi-autonomous AI agents. This is the pivotal dimension for determining the long-term viability of development methodologies in an agentic future.

Research in human–AI teaming (Seeber, 2020; Shneiderman, 2020), distributed cognition (Hutchins, 1995), and AI-assisted engineering (Qian et al., 2023; Chen, 2021; Bavishi, 2022) demonstrates that human–AI collectives operate in ways that differ fundamentally from traditional human teams. Key properties include:

- coordination costs approaching **zero**;
- shared state becoming **virtualized** rather than socially negotiated;
- execution becoming **automated** and **parallelized**;
- discovery and refinement loops becoming **continuous**;
- human responsibility shifting toward **intent, oversight, constraints, and ethics**.

Frameworks predicated on human social behaviors, interpersonal synchronization, fixed role structures, or cognitive limitations generally fail this dimension, as such assumptions do not transfer to ecosystems where AI agents perform the majority of execution while humans focus on higher-order decision-making and interpretation.

### A.10.1 BINARY SCORING TABLE FOR D6

Framework	Q1	Q2	Q3	Q4	Q5
OBAF	1	1	1	1	1
Lean Startup	1	1	1	1	0
DevOps/SRE	1	1	1	1	1
Kanban	1	1	1	1	1
Extreme Programming (XP)	0	0	0	1	0
Scrum	0	0	0	0	0
LeSS	0	0	0	0	0
SAFe	0	0	0	0	0
Waterfall / V-Model / PRINCE2	0	0	0	0	0

**Table 7:** Binary scoring for D6: Human–AI Ecosystem Compatibility. Q1–Q5 are defined in Section A.3.

### A.10.2 RATIONALE FOR EACH FRAMEWORK

**OBAF.** OBAF satisfies all D6 indicators. It is fundamentally agnostic to the nature of the team and instead centers on outcomes, constraints, hypothesis-driven experimentation, and evidence. None of its principles depend on human–human interaction (Q1=1, Q2=1). Planning is recognitional (NDM), which integrates naturally with AI-generated options (Q3=1). Its coordination model is based on intent rather than interpersonal negotiation (Q4=1). OBAF’s cognitive model (signals, hypotheses, learning loops) is compatible with AI augmentation (Q5=1), making it uniquely aligned with human–AI ecosystems.

**LEAN STARTUP.** Lean Startup satisfies Q1–Q4 because its build–measure–learn cycle can be executed with humans, AI agents, or both. Discovery and execution can be automated, and no human roles or interpersonal dependencies are required. However, Q5=0: Lean Startup relies on human-driven interpretation of customer learning and hypothesis framing. While AI can support this process, Lean’s cognitive model is not fully AI-augmented without adjustments.

**DEVOPS/SRE.** DevOps scores 5/5. Modern DevOps pipelines rely heavily on automation, telemetry, and continuous verification, which map well to agent-driven execution. SRE roles are increasingly augmented by AI-based alerting, anomaly detection, and auto-remediation (Forsgren, Humble, and Kim, 2018). DevOps coordination mechanisms generalize seamlessly to agents (Q4=1), and cognitive load is reduced by automation, satisfying Q5=1.

**KANBAN.** Kanban also scores 5/5. Kanban boards and workflow limits do not rely on human-specific behaviors (Q1–Q3=1). Kanban’s model of coordination through visualization (distributed cognition) extends naturally to human–AI collectives. AI agents can autonomously pull tasks, update board states, and manage flow constraints with high fidelity. AI augmentation further reduces the cognitive overhead of maintaining work-in-progress limits (Q5=1).

**XP (EXTREME PROGRAMMING).** XP scores only 1/5. It fails Q1–Q3 because its core practices (pair programming, collective ownership, customer-on-site, standups) are explicitly designed around human interaction, interpersonal feedback, and team-level cognition (Beck, 1999). XP’s coordination model depends heavily on synchronous collaboration. Q4=1 because planning models (small batch work, continuous integration) can be extended to agents. Q5=0 because XP assumes human cognition and human feedback as its basis.

**SCRUM.** Scrum receives 0 for all indicators. Its identity is tied to human team structures, prescribed roles tied to human psychology (PO, SM), interpersonal synchronization events, and human-driven decomposition of work. None of these generalize to agentic ecosystems. Coordination models (sprints, reviews, retrospectives) presuppose human time budgets and cognitive patterns. Research shows Scrum’s success relies on human cohesion and communication (Kuusinen et al., 2017), making it fundamentally incompatible.

**LESS.** LeSS inherits all of Scrum’s human-centric assumptions and expands them into multi-team networks. It depends on social coordination at scale, cross-team retrospectives, and multi-team refinement. LeSS assumes human cognition and human negotiation at all levels and therefore fails all indicators.

**SAFE.** SAFe is even more deeply rooted in human coordination structures than Scrum or LeSS. It requires formal hierarchies, cross-team and cross-role alignment, human-driven prioritization, and construct such as ARTs (Agile Release Trains), which have no analog in human–AI ecosystems. All indicators fail.

**WATERFALL / V-MODEL / PRINCE2.** Traditional plan-driven methodologies rely entirely on sequential human-driven analysis, design, verification, and signoff. None of their structures generalize to AI agents. The V-model assumes human-driven decomposition and human-driven verification stages. PRINCE2 depends heavily on human decision gates and role responsibilities. All indicators fail.

### A.10.3 DIMENSION SUMMARY

D6 clearly distinguishes between frameworks that treat “the team” as a human social unit (Scrum, LeSS, SAFe, XP, Waterfall) and those that treat development as a flow of hypotheses, experiments, and evidence (OBAF, Lean Startup, DevOps, Kanban). Frameworks in the latter category perform significantly better in human–AI ecosystems, where human cognitive roles shift from task execution to intent-setting, constraint definition, ethical oversight, and interpretation of ambiguous signals.

OBAF is uniquely strong because it was never tied to team rituals; its cognitive architecture (outcomes, constraints, recognitional planning, evidence) directly matches how human–AI ecosystems function in practice. This explains its exceptionally high compatibility score in the main analysis.

## A.11 FINAL COMPATIBILITY RESULTS

Using the scoring model defined in Sections A.3–A.4 and the binary scores assigned in Sections A.5–A.10, each framework’s compatibility percentage is computed by averaging its six dimension scores and normalizing to a 0–100 scale.

Framework	Compatibility (%)
Outcome-Based Agile (OBAF)	100%
Lean Startup	93%
DevOps / SRE	80%
Kanban	77%
Extreme Programming (XP)	56%
Scrum	22%
LeSS	16%
SAFe	8%
Waterfall / V-Model / PRINCE2	6%

**Table 8:** Final compatibility estimates for each framework, computed from the structured multi-dimensional model.

These results indicate strong alignment between outcome- and discovery-oriented frameworks (OBAF, Lean Startup, DevOps) and human–AI development ecosystems. In contrast, frameworks structured around human coordination and ritualized processes (Scrum, LeSS, SAFe, and Waterfall) exhibit structural incompatibility with environments characterized by low-cost iteration and autonomous agent execution.

## A.12 METHODOLOGICAL VALIDITY

The methodology used in this appendix belongs to a class of scientifically recognized techniques for evaluating complex socio-technical systems under uncertainty. The approach draws from:

1. **Structured Expert Judgment (SEJ)** as formalized by Cooke (Cooke, 1991), which relies on transparent decomposition of complex evaluations into smaller, objectively assessed indicators.
2. **Multi-Criteria Decision Analysis (MCDA)**, widely used in engineering and decision theory to compare alternatives across heterogeneous criteria.
3. **Architectural evaluation frameworks** such as ATAM, which evaluate software architectures via structured qualitative reasoning rather than empirical measurement.
4. **Sociotechnical systems analysis**, which assesses the alignment between organizational structures and technical systems (Trist, 1981; Sailer and McCulloh, 2020).
5. **Empirical Software Engineering**, which frequently uses systematic qualitative scoring when empirical measures are infeasible (e.g., future-state analysis, methodological compatibility studies).

The model is scientifically valid because:

- Indicators are derived from empirical literature and theory.
- All scoring criteria are binary (reducing arbitrariness).
- Every score has a documented rationale, ensuring transparency.
- The aggregation function (mean) is simple, monotonic, and interpretable.
- No indicator has implicit weighting unless justified theoretically.
- Dimensional independence is reasonable given the conceptual boundaries of D1–D6.

This makes the model reproducible, falsifiable, and analyzable—meeting the standards of academic rigor for conceptual comparative work.

## A.13 LIMITATIONS

Despite the rigor, several limitations must be acknowledged:

### LACK OF EMPIRICAL FUTURE-STATE DATA.

There is no large-scale empirical data on fully agentic human–AI software development ecosystems. Therefore, the model evaluates structural compatibility, not measured performance.

### EXPERT-JUDGMENT SUBJECTIVITY.

Although binary scoring reduces subjectivity, expert interpretation remains a factor. However, the transparency of the scoring allows replication and critique.

### DEPENDENCE ON CURRENT LITERATURE.

The model relies on contemporary research in Agile, DevOps, and AI-assisted development, which itself is evolving.

### FRAMEWORK INTERPRETATION VARIANCE.

Frameworks such as Scrum or XP are sometimes implemented in modified forms; this analysis scores the canonical definitions.

### SIMPLIFIED DIMENSIONAL INDEPENDENCE.

We treat D1–D6 as independent for purposes of scoring, though in practice interaction effects may exist (e.g., ritual dependence often correlates with coordination dependence).

These limitations are inherent in conceptual comparative work but do not invalidate the structure or utility of the model.

## A.14 REPRODUCIBILITY AND INTER-RATER RELIABILITY

Reproducibility is ensured through:

- Complete disclosure of indicators (Section A.3)
- Full scoring tables (Sections A.5–A.10)
- Detailed rationales (Sections A.5–A.10)

Any researcher can independently rescore the frameworks using:

1. the same binary indicators,
2. the same definitions,
3. the same canonical descriptions of the frameworks.

Inter-rater reliability can be improved through:

- Multiple expert scorers,
- Pairwise adjudication,
- Calibration sessions using sample frameworks,
- Sensitivity analysis (e.g., adjusting indicators).

These practices align with established SEJ protocols (Cooke, 1991; Aspinall, 2010).

## A.15 CONCLUSION

The compatibility model presented here provides a scientifically grounded, transparent, and reproducible framework for evaluating the suitability of software development methodologies in emerging human–AI ecosystems. It makes explicit the structural assumptions embedded within traditional frameworks and clarifies why outcome-oriented, discovery-driven, and coordination-minimal methods exhibit the highest compatibility with agent-augmented development environments.

OBAT’s perfect compatibility score is not the result of bias but of structural alignment between its core epistemic principles and the demands of human–AI development: continuous discovery, constraint framing, recognitional planning, and evidence-driven iteration. As agentic toolchains become increasingly prevalent, such frameworks will form the epistemic and governance backbone of future software engineering practice.

## APPENDIX B: ORGANIZATIONAL STRUCTURES IN HUMAN–AI SOFTWARE ECOSYSTEMS

### B.1 INTRODUCTION

Appendix A evaluated delivery frameworks (e.g., OBAF, Lean Startup, DevOps/SRE, Scrum, XP) using six epistemic dimensions (D1–D6) grounded in research on naturalistic decision-making, sociotechnical systems, coordination theory, and human–AI teaming. These dimensions assessed how well each method remains viable in a software development environment transformed by autonomous coding agents, cheap iteration, continuous experimentation, and hybrid human–AI cognitive loops.

Delivery frameworks, however, operate within broader organizational structures. These structures define persistent boundaries, coordination mechanisms, decision rights, incentives, communication pathways, and the distribution of cognitive load. Organizational architecture therefore shapes—and frequently constrains—what teams, and increasingly hybrid human–AI units, can accomplish. Delivery methods determine how work is performed; organizational structures determine the conditions under which such work can succeed.

Organizational structures do not iterate, discover, or deliver software directly. Instead, they create sociotechnical environments that enable—or obstruct—autonomous decision-making, rapid learning, local experimentation, architectural evolution, and outcome accountability. In the emerging agentic development paradigm, where coordination costs collapse, discovery becomes continuous, and AI systems take on execution and analysis tasks, certain organizational forms naturally support high-autonomy, outcome-driven work, while others introduce friction, bottlenecks, or misalignment.

The purpose of this appendix is to evaluate a broad range of organizational structures—classical, modern, sociotechnical, and post-bureaucratic—against the structural requirements of human–AI software development ecosystems. These requirements include alignment of organizational and architectural boundaries, distribution of decision rights to autonomous units, continuous structural adaptability, integration of discovery with delivery, and generalization of coordination patterns beyond human-only negotiation or hierarchical control.

Because organizational affordances rarely exist in purely present-or-absent form, this appendix employs a ternary scoring model. Each indicator  $q_{k,i}(M)$  takes one of three values:

$$q_{k,i}(M) \in \{0, 0.5, 1\},$$

where 1 denotes strong structural support, 0.5 indicates partial or conditional support, and 0 indicates

structural misalignment or obstruction. This approach aligns with established practices in organizational research, which commonly rely on graded assessments, maturity scales, and fit-gap analysis rather than binary classification.

Twenty organizational models are evaluated using this approach, spanning industrial-era designs (Functional Hierarchy, Matrix Organization, Project-Based Organization, Capability-Based Silos), modern product and platform structures (Divisional Structure, Lean Enterprise (Humble, Molesky, and O'Reilly, 2015), Spotify Model, Team Topologies (Skelton and Pais, 2019), Platform Organizations), self-management and socio-technical approaches (Rendanheyi (Wu et al., 2021), Sociotechnical Systems (Trist, 1981), Holacracy (Robertson, 2015), Sociocracy 3.0, Teal Organizations (Laloux, 2014)), and analytic or theoretical models (Leavitt's Diamond, Galbraith's Information-Processing View (Galbraith, 1973), Contingency Theory (Burns and Stalker, 1961), The Fifth Discipline (Senge, 1990), Theory of Constraints (Goldratt and Cox, 1984), and OKR-driven organizations).

The resulting compatibility scores reflect how well each organizational structure aligns with the structural requirements of the centaur-era, human–AI software development reality. They do not measure compatibility with any particular delivery framework. A later section (B.12) provides a qualitative analysis of how these organizational models pair with OBAF and Lean Startup, since compatibility with the human–AI development paradigm does not automatically imply compatibility with specific delivery methodologies.

### B.2 THEORETICAL FOUNDATIONS

The six organizational dimensions used in this appendix (O1–O6) draw on well-established research areas that explain why some structures thrive in dynamic, high-learning environments while others fail.

**ORGANIZATIONAL STRUCTURE AND DESIGN.** Classic studies demonstrate that organizational form must reflect environmental conditions. Chandler's historical analysis showed that structure follows strategy and that divisional structures improve responsiveness under diversification (Chandler, 1962). Burns and Stalker distinguished mechanistic structures, suited for stable conditions, from organic structures, suited for dynamic ones (Burns and Stalker, 1961). Mintzberg identified structural configurations and their embedded coordination mechanisms (Mintzberg, 1979), while Galbraith framed organizations as information-processing systems whose structures must align with the level of uncertainty and coordination demand (Galbraith, 1973).

**SOCIOTECHNICAL SYSTEMS THEORY.** STS research conceptualizes organizations as jointly optimized social and technical systems (Trist, 1981). It emphasizes autonomous work groups, minimal hand-offs, and dynamic boundary-setting—all structurally aligned with high-flow, AI-augmented software development.

### COORDINATION THEORY AND COGNITIVE LOAD.

Coordination overhead is a dominant factor in software engineering performance (Brooks, 1995). Structures that demand heavy cross-functional synchronization introduce delays and cognitive burden. Modern research on team boundaries and cognitive load (Skelton and Pais, 2019) reinforces that organizational design must minimize unnecessary coordination and support high-autonomy flow.

### HUMAN–AI TEAMING AND DISTRIBUTED COGNITION.

In hybrid teams, cognitive work is distributed across humans and AI agents, reducing reliance on interpersonal negotiation and enabling automated execution and analysis (Seeber et al., 2020; Hutchins, 1995). Organizations dependent on human-centric approval chains or social hierarchy become mismatched in such ecosystems.

### LEAN ENTERPRISE AND CONTINUOUS IMPROVEMENT.

Research in Lean Thinking (Womack and Jones, 1996), Toyota Kata (Rother, 2010), and Lean Enterprise transformations (Humble, Molesky, and O'Reilly, 2015) demonstrates that decentralized decision-making, small-batch learning, fast feedback loops, and continuous improvement increase adaptability and performance. These principles are corroborated by large-scale DevOps studies (Forsgren, Humble, and Kim, 2018), which show that outcome-based, feedback-rich, autonomy-supporting structures enable high-velocity digital delivery.

These research foundations motivate the six organizational dimensions introduced in Section B.3 (O1–O6). Together, they provide a structured basis for evaluating whether an organizational model supports the conditions necessary for outcome-driven, AI-augmented software development in the centaur era.

## B.3 EVALUATION DIMENSIONS AND INDICATORS

Each organizational model  $M$  is evaluated along six structural dimensions  $O_1, \dots, O_6$ . Each dimension contains five indicators  $q_{k,i}(M)$ , scored using a ternary scale:

$$q_{k,i}(M) \in \{0, 0.5, 1\},$$

where:

- **1** indicates strong structural support for the indicator,
- **0.5** indicates partial or conditional support,
- **0** indicates structural misalignment or obstruction.

This ternary scoring approach reflects the nature of organizational structures, which rarely express properties in binary form. It aligns with established practices in organizational research, where graded scales and maturity models are commonly used to capture partial fit, conditional support, or context-dependent behavior (Mintzberg, 1979; Galbraith, 1973; Burns and Stalker, 1961; Trist, 1981).

The six dimensions evaluate whether an organizational structure provides the enabling conditions for AI-augmented, outcome-driven, high-autonomy delivery in human–AI ecosystems.

### O1: BOUNDARY–OUTCOME ALIGNMENT

This dimension evaluates whether organizational boundaries align with value streams, coherent domains, and end-to-end outcomes. Structural alignment between organizational units and system architecture reduces handoffs, minimizes coordination burden, and supports fast flow (Conway, 1968b; Skelton and Pais, 2019).

1. **O1-Q1:** Organizational units map to value streams, products, or cohesive domains. (0 = functional or project boundaries; 0.5 = partial mapping; 1 = clear domain/value-stream mapping)
2. **O1-Q2:** Boundaries minimize cross-unit handoffs for value creation. (0 = many handoffs; 0.5 = some; 1 = minimal or none)
3. **O1-Q3:** Organizational boundaries align with architectural seams (Conway alignment). (0 = misaligned; 0.5 = mixed; 1 = strongly aligned)
4. **O1-Q4:** Accountability is defined in terms of outcomes rather than internal functions. (0 = tasks/outputs; 0.5 = mixed; 1 = clear outcome ownership)
5. **O1-Q5:** Outcome responsibility remains stable even as execution strategies change. (0 = responsibilities shift frequently; 0.5 = partially stable; 1 = stable outcome ownership)

## O2: AUTONOMY & DECISION RIGHTS

This dimension evaluates whether organizational units have the autonomy and decision rights required for high-velocity, AI-augmented environments. Decentralized authority improves performance in uncertain and dynamic contexts (Burns and Stalker, 1961; Robertson, 2015).

1. **O2-Q1:** Decision authority resides close to the work and outcome. (0 = centralized; 0.5 = mixed; 1 = decentralized)
2. **O2-Q2:** The structure does not rely on hierarchical approval flows. (0 = strong hierarchy; 0.5 = partial; 1 = minimal hierarchy)
3. **O2-Q3:** Units can act independently without cross-department coordination rituals. (0 = coordination-heavy; 0.5 = moderate; 1 = independent)
4. **O2-Q4:** The structure minimizes dependence on human-only coordination roles (project managers, liaisons, committees). (0 = heavy reliance; 0.5 = partial; 1 = minimal reliance)
5. **O2-Q5:** The structure scales down coherently to small (1–2 person) autonomous units. (0 = impossible; 0.5 = difficult; 1 = natural fit)

## O3: STRUCTURAL ADAPTABILITY

This dimension evaluates whether the organizational structure can be reconfigured rapidly and at low cost as new learning emerges. Adaptive structures outperform rigid forms in dynamic, uncertain environments (Trist, 1981; Galbraith, 1973).

1. **O3-Q1:** Structural changes do not require long or rigid planning cycles. (0 = long cycles; 0.5 = moderate; 1 = short or continuous)
2. **O3-Q2:** No phase-gated governance (steering committees, sequential lifecycles). (0 = strong; 0.5 = partial; 1 = none)
3. **O3-Q3:** Units can be reconfigured or retargeted with minimal organizational overhead. (0 = costly/slow; 0.5 = mixed; 1 = low-cost)
4. **O3-Q4:** Boundaries and interfaces can be revised as learning emerges. (0 = rigid; 0.5 = occasionally flexible; 1 = fluid)
5. **O3-Q5:** Continuous structural adaptation is expected and supported. (0 = rare; 0.5 = episodic; 1 = continuous norm)

## O4: OUTCOME ACCOUNTABILITY & VALUE FLOW

This dimension evaluates whether units are accountable for end-to-end outcomes, integrate quality attributes, and use empirical signals to drive decisions. High-performing digital organizations rely on outcome accountability rather than output compliance (Forsgren, Humble, and Kim, 2018).

1. **O4-Q1:** Units hold end-to-end responsibility for measurable outcomes. (0 = none; 0.5 = partial; 1 = explicit)
2. **O4-Q2:** Success is not defined by deliverables or activity levels. (0 = deliverable-driven; 0.5 = mixed; 1 = outcome-driven)
3. **O4-Q3:** Quality attributes (reliability, performance, safety) are structurally embedded. (0 = absent; 0.5 = partial; 1 = embedded)
4. **O4-Q4:** Empirical performance signals (telemetry, metrics) drive decisions. (0 = weak; 0.5 = partial; 1 = strong)
5. **O4-Q5:** Outcome responsibilities remain stable even when execution strategies vary. (0 = shifting; 0.5 = semi-stable; 1 = stable)

## O5: DISCOVERY INTEGRATION & LEARNABILITY

This dimension evaluates whether discovery—research, experimentation, telemetry, hypothesis testing—is structurally embedded near where work occurs. Modern product and platform organizations integrate discovery with delivery to enable rapid feedback and learning (Forsgren, Humble, and Kim, 2018; Womack and Jones, 1996; Rother, 2010).

1. **O5-Q1:** Research and learning functions are not isolated into upstream departments. (0 = isolated; 0.5 = partly integrated; 1 = fully integrated)
2. **O5-Q2:** Units can run experiments autonomously without central approval. (0 = no; 0.5 = limited; 1 = yes)
3. **O5-Q3:** Discovery is integrated with delivery rather than episodic or upstream. (0 = separated; 0.5 = partially; 1 = fully)
4. **O5-Q4:** The structure supports automated analysis, AI-based insights, and rapid experimentation. (0 = little support; 0.5 = partial; 1 = strong support)
5. **O5-Q5:** Decision processes prioritize empirical evidence over hierarchy or ritual. (0 = weak; 0.5 = mixed; 1 = evidence-driven)

## O6: HUMAN–AI COORDINATION COMPATIBILITY

This dimension evaluates whether organizational coordination mechanisms generalize beyond human-only processes. Research shows that AI agents reduce interpersonal negotiation, virtualize shared state, and enable automated coordination and monitoring (Seeger et al., 2020; Hutchins, 1995).

1. **O6-Q1:** Coordination does not depend on human social hierarchy or interpersonal negotiation. (0 = dependent; 0.5 = mixed; 1 = independent)
2. **O6-Q2:** Roles are defined by responsibilities rather than human-specific attributes. (0 = human-centric; 0.5 = mixed; 1 = abstract/responsibility-based)
3. **O6-Q3:** Task decomposition is not tied to strictly human labor categories. (0 = human-labor-based; 0.5 = partial; 1 = agent-generalizable)
4. **O6-Q4:** Coordination patterns can be represented as interfaces/contracts suitable for AI orchestration. (0 = no; 0.5 = partial; 1 = yes)
5. **O6-Q5:** Human cognitive load is concentrated on intent, oversight, and ethics—amenable to automation elsewhere. (0 = diffuse; 0.5 = partial; 1 = strongly concentrated)

## B.4 SCORING FORMULA

Each organizational model  $M$  is evaluated using ternary indicators  $q_{k,i}(M) \in \{0, 0.5, 1\}$ , where  $k \in \{1, \dots, 6\}$  indexes the six organizational dimensions (O1–O6), and  $i \in \{1, \dots, 5\}$  indexes the indicators within each dimension.

A value of:

- **1** indicates strong structural support for the indicator,
- **0.5** indicates partial, conditional, or incomplete support,
- **0** indicates structural misalignment or obstruction.

The dimension score for  $O_k$  is computed as the arithmetic mean of its five indicators:

$$O_k(M) = \frac{1}{5} \sum_{i=1}^5 q_{k,i}(M).$$

The overall compatibility score for organizational model  $M$  is the mean of its six dimension scores:

$$\text{Compat}(M) = \frac{1}{6} \sum_{k=1}^6 O_k(M).$$

Finally, a normalized compatibility percentage is computed as:

$$\text{Compat\%}(M) = 100 \cdot \text{Compat}(M).$$

This ternary scoring model reflects the fact that organizational affordances are rarely binary in nature. Structures may partially support autonomy, boundary alignment, discovery integration, or AI-compatible coordination without fully embodying these properties. The ternary scale enables a more nuanced and academically defensible evaluation of structural fit for human–AI (centaur) software development ecosystems.

The following sections (B.5–B.10) present the scoring tables and indicator-by-indicator rationales for each organizational model under dimensions O1–O6.

## B.5 DIMENSION O1: BOUNDARY–OUTCOME ALIGNMENT

This dimension evaluates whether an organizational structure aligns its boundaries with value streams, coherent domains, and end-to-end outcomes. Empirical and theoretical work in organizational design shows that structural boundaries heavily influence coordination cost, cognitive load, and system architecture (Chandler, 1962; Mintzberg, 1979; Galbraith, 1973). In software-intensive organizations, Conway’s Law further predicts that system architecture will mirror organizational communication patterns (Conway, 1968b). Structures that align units to value streams or domains and assign clear outcome responsibility tend to support high-flow, outcome-based work (Skelton and Pais, 2019).

### B.5.1 TERNARY SCORING TABLE FOR O1

Organizational Model	Q1	Q2	Q3	Q4	Q5
Functional Hierarchy	0	0	0	0	0
Matrix Organization	0	0	0	0	0
Project-Based Organization	0	0	0	0	0
Capability-Based (Horizontal Silos)	0	0	0	0	0
Divisional (M-form)	1	0.5	0.5	1	1
Lean Enterprise	0.5	0.5	0.5	1	1
Spotify Model	1	0.5	0.5	1	1
Team Topologies	1	1	1	1	1
Platform Organization Model	1	0.5	1	1	1
Rendanheyi (Haier)	1	1	1	1	1
Sociotechnical Systems (STS)	1	1	0.5	1	1
Leavitt’s Diamond	0.5	0.5	0.5	0.5	0.5
Galbraith’s Information-Processing Model	0.5	0.5	0.5	1	0.5
Contingency Theory	0.5	0.5	0.5	1	0.5
Teal Organizations	0.5	0.5	0.5	1	0.5
Holacracy	1	0.5	0.5	1	1
Sociocracy 3.0	0.5	0.5	0.5	1	0.5
The Fifth Discipline (Learning Organization)	0.5	0.5	0.5	0.5	0.5
OKR-Driven Organizations	0	0.5	0	1	0.5
Theory of Constraints (as Org Model)	0.5	0.5	0.5	1	1

**Table 9:** Ternary scoring for O1: Boundary–Outcome Alignment. Indicators O1–Q1–O1–Q5 are defined in Section B.3.

### B.5.2 RATIONALE FOR EACH ORGANIZATIONAL MODEL

**FUNCTIONAL HIERARCHY.** Q1–Q5 = 0. Units are defined by internal functions (e.g., Development, QA, Operations); boundaries do not map to value streams or domains, create many handoffs, misalign with architecture, and provide no stable outcome ownership (Weber, 1978).

**MATRIX ORGANIZATION.** Q1–Q5 = 0. Boundaries are drawn along two orthogonal axes (function and project) rather than value streams. Accountability is fragmented and outcome responsibilities shift with changing project portfolios (Galbraith, 1973).

**PROJECT-BASED ORGANIZATION.** Q1–Q5 = 0. Projects are temporary structures defined by deliverables and timelines, not domains or enduring outcomes (Kerzner, 2017). Handoffs are common, and outcome ownership dissolves when projects end.

**CAPABILITY-BASED (HORIZONTAL SILOS).** Q1–Q5 = 0. Capability silos (e.g., “data team,” “infra team”) optimize internal specialties rather than value streams. Boundaries are misaligned with products and lead to constant cross-silo coordination.

**DIVISIONAL (M-FORM).** Q1=1, Q2=0.5, Q3=0.5, Q4=1, Q5=1. Divisions are typically organized by product, market, or geography (Chandler, 1962), providing clear outcome responsibility (Q1, Q4, Q5=1). However, shared services and legacy systems often create handoffs and partial architectural misalignment (Q2, Q3=0.5).

**LEAN ENTERPRISE.** Q1=0.5, Q2=0.5, Q3=0.5, Q4=1, Q5=1. Lean Enterprise transformations aim to align structures to value streams and outcomes (Humble, Molesky, and O'Reilly, 2015; Forsgren, Humble, and Kim, 2018), but often overlay existing functional/matrix forms. In practice this yields partial mapping and reduced, but not eliminated, handoffs and misalignment (0.5 scores for Q1–Q3).

**SPOTIFY MODEL.** Q1=1, Q2=0.5, Q3=0.5, Q4=1, Q5=1. Squads and tribes are aligned to product areas, with clear outcome attachments and stable responsibilities. However, chapters and guilds introduce cross-cutting dependencies, and architectural alignment is partial and practice-dependent.

**TEAM TOPOLOGIES.** Q1–Q5 = 1. Stream-aligned teams map directly to value streams; enabling, platform, and complicated-subsystem teams are defined to support those streams (Skelton and Pais, 2019). Boundaries deliberately follow architectural seams and outcome responsibilities are stable.

**PLATFORM ORGANIZATION MODEL.** Q1=1, Q2=0.5, Q3=1, Q4=1, Q5=1. Product teams own external outcomes and platform teams own internal service outcomes. Interfaces align strongly with architecture, but some handoffs to platform are inherent (Q2=0.5).

**RENDANHEYI (HAIER).** Q1–Q5 = 1. Micro-enterprises are designed around customer and market outcomes (Wu et al., 2021). Units have stable outcome accountability, boundaries are aligned to value, and coordination occurs through contracts and platform interfaces.

**SOCIOTECHNICAL SYSTEMS (STS).** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=1. STS designs autonomous work groups around complete work systems (Trist, 1981), reducing handoffs and defining clear responsibilities. Architectural alignment is implicit rather than explicit (Q3=0.5).

**LEAVITT'S DIAMOND.** Q1–Q5 = 0.5. Leavitt's model describes mutual adjustment of tasks, people, structure, and technology. It supports value-aligned boundaries conceptually, but does not prescribe specific patterns, resulting in partial support on all indicators.

**GALBRAITH'S INFORMATION-PROCESSING MODEL.** Q1=0.5, Q2=0.5, Q3=0.5, Q4=1, Q5=0.5. Galbraith emphasizes structural alignment to information-processing demands and performance outcomes (Galbraith, 1973), supporting outcome accountability (Q4=1) and partial alignment on other indicators.

**CONTINGENCY THEORY.** Q1=0.5, Q2=0.5, Q3=0.5, Q4=1, Q5=0.5. Contingency theory argues that effective organizations adapt structure to environmental demands, including outcome focus (Burns and Stalker, 1961). It supports outcome accountability but does not prescribe specific boundary patterns.

**TEAL ORGANIZATIONS.** Q1=0.5, Q2=0.5, Q3=0.5, Q4=1, Q5=0.5. Teal structures organize around evolutionary purpose and value contribution (Laloux, 2014), but boundaries can be fluid and informal. This yields partial boundary– outcome alignment with strong purpose-based accountability.

**HOLACRACY.** Q1=1, Q2=0.5, Q3=0.5, Q4=1, Q5=1. Holacracy defines roles and circles with explicit domains and accountabilities (Robertson, 2015), strongly supporting stable outcome ownership. Architectural alignment and handoff minimization are possible but depend on implementation (Q2, Q3=0.5).

**SOCIOCRACY 3.0.** Q1=0.5, Q2=0.5, Q3=0.5, Q4=1, Q5=0.5. Sociocracy organizes around domains and consent-based circles, with clear purpose per circle. Outcome accountability is strong (Q4=1), while boundary–architecture alignment is partially supported.

**THE FIFTH DISCIPLINE (LEARNING ORGANIZATION).**  $Q1=Q5 = 0.5$ . Learning organizations emphasize systems thinking and aligning structures to value flows (Senge, 1990), but provide conceptual rather than prescriptive guidance on boundary design. This yields consistent partial support.

**OKR-DRIVEN ORGANIZATIONS.**  $Q1=0, Q2=0.5, Q3=0, Q4=1, Q5=0.5$ . OKRs overlay existing structures and define outcome objectives within those structures. They improve outcome accountability ( $Q4=1$ ) and partially stabilize responsibilities ( $Q2, Q5=0.5$ ), but typically do not realign boundaries or architecture ( $Q1, Q3=0$ ).

**THEORY OF CONSTRAINTS (TOC).**  $Q1=0.5, Q2=0.5, Q3=0.5, Q4=1, Q5=1$ . TOC focuses on system-level flow and bottleneck management (Goldratt and Cox, 1984). It encourages organizing around constraints and system outcomes, but often as an overlay on existing structures, leading to partial boundary alignment and strong outcome accountability.

### B.5.3 DIMENSION SUMMARY

O1 reveals a clear split between traditional function- and project-oriented structures (Functional, Matrix, Project, Capability Silos), which provide no meaningful boundary–outcome alignment, and modern or analytic structures that align units more tightly to value streams, domains, or system outcomes (Divisional, Lean Enterprise, Team Topologies, Platform, Rendanheyi, STS). Lean Enterprise and Spotify sit between these extremes, improving alignment on top of legacy forms. Conceptual models such as Leavitt's Diamond, Contingency Theory, and the Learning Organization offer partial guidance but require concrete instantiation to achieve strong alignment. OKR overlays improve outcome accountability but do not, by themselves, repair misaligned boundaries.

## B.6 DIMENSION O2: AUTONOMY & DECISION RIGHTS

This dimension evaluates whether organizational units possess the autonomy and decision rights required for high-velocity, AI-augmented environments. Research in organizational design shows that decentralized authority, local control, and minimized coordination overhead strongly correlate with adaptability and innovation in dynamic contexts (Burns and Stalker, 1961). Lean Enterprise transformations (Humble, Molesky, and O'Reilly, 2015) and modern product-centric organizations emphasize these principles, while traditional functional or matrix structures restrict local decision-making through hierarchical control or dual reporting lines.

### B.6.1 TERNARY SCORING TABLE FOR O2

Organizational Model	Q1	Q2	Q3	Q4	Q5
Functional Hierarchy	0	0	0	0	0
Matrix Organization	0	0	0	0	0
Project-Based Organization	0	0	0	0	0
Capability-Based (Horizontal Silos)	0	0	0	0	0
Divisional (M-form)	1	0.5	0.5	0.5	1
Lean Enterprise	1	1	0.5	0.5	1
Spotify Model	1	1	1	0.5	1
Team Topologies	1	1	1	1	1
Platform Organization Model	1	1	1	1	1
Rendantheyi (Haier)	1	1	1	1	1
Sociotechnical Systems (STS)	1	1	1	1	1
Leavitt's Diamond	0.5	0.5	0.5	0.5	0.5
Galbraith's Information-Processing Model	1	1	1	1	1
Contingency Theory	1	1	0.5	0.5	0.5
Teal Organizations	1	1	1	1	1
Holacracy	1	1	1	1	1
Sociocracy 3.0	1	1	1	1	1
The Fifth Discipline (Learning Organization)	0.5	1	0.5	0.5	0.5
OKR-Driven Organizations	0.5	0	0	0	0.5
Theory of Constraints (as Org Model)	1	1	1	1	1

**Table 10:** Ternary scoring for O2: Autonomy & Decision Rights. Indicators O2-Q1–O2-Q5 are defined in Section B.3.

### B.6.2 RATIONALE FOR EACH ORGANIZATIONAL MODEL

**FUNCTIONAL HIERARCHY.** Q1–Q5 = 0. Decision rights are centralized at managerial layers; units cannot act independently and rely on hierarchical approval at every stage (Weber, 1978). No aspect of the structure supports autonomous work.

**MATRIX ORGANIZATION.** Q1–Q5 = 0. Dual reporting splits authority between functional and project managers. Decisions require human negotiation, preventing autonomy and delaying action (Galbraith, 1973).

**PROJECT-BASED ORGANIZATION.** Q1–Q5 = 0. Projects are run through steering bodies, project managers, and executive approvals (Kerzner, 2017). Local autonomy is minimal.

**CAPABILITY-BASED (HORIZONTAL SILOS).** Q1–Q5 = 0. Silos depend on upstream/downstream handoffs; units cannot deliver or decide independently.

**DIVISIONAL (M-FORM).** Q1=1, Q2=0.5, Q3=0.5, Q4=0.5, Q5=1. Divisions have strong local decision rights and act autonomously (Chandler, 1962). However, shared services and corporate governance reduce full autonomy in practice.

**LEAN ENTERPRISE.** Q1=1, Q2=1, Q3=0.5, Q4=0.5, Q5=1. Lean Enterprise promotes decentralized authority, empowered teams, and local decision-making (Humble, Molesky, and O'Reilly, 2015). However, transformation overlays often retain coordination roles, yielding partial independence (Q3–Q4).

**SPOTIFY MODEL.** Q1=1, Q2=1, Q3=1, Q4=0.5, Q5=1. Squads act as mini-startups with substantial autonomy. Chapters and tribes introduce light coordination, keeping Q4 at 0.5.

**TEAM TOPOLOGIES.** Q1–Q5 = 1. Stream-aligned teams are designed for autonomy, with minimal coordination roles and clear decision rights (Skelton and Pais, 2019).

**PLATFORM ORGANIZATION MODEL.** Q1–Q5 = 1. Product and platform teams have autonomous execution domains; coordination occurs via contracts, not managers.

**RENDANHEYI (HAIER).** Q1–Q5 = 1. Micro-enterprises are fully autonomous business units with market-facing authority (Wu et al., 2021). Decision rights are decentralized and explicit.

**SOCIOTECHNICAL SYSTEMS (STS).** Q1–Q5 = 1. STS designs autonomous work groups with full local control over workflows, decisions, and value delivery (Trist, 1981).

**LEAVITT'S DIAMOND.** Q1–Q5 = 0.5. The model suggests decentralized adaptation but does not define explicit mechanisms for autonomy, leading to partial support.

**GALBRAITH'S INFORMATION-PROCESSING MODEL.** Q1–Q5 = 1. Under high uncertainty, Galbraith recommends decentralizing authority to the nodes closest to information (Galbraith, 1973). Structure is explicitly autonomy-supportive.

**CONTINGENCY THEORY.** Q1=1, Q2=1, Q3=0.5, Q4=0.5, Q5=0.5. In uncertain contexts, autonomy is prescribed; in stable ones, centralization is sometimes appropriate. Thus mixed scores (0.5).

**TEAL ORGANIZATIONS.** Q1–Q5 = 1. Teal principles emphasize self-management and distributed authority (Laloux, 2014). Units act autonomously by design.

**HOLACRACY.** Q1–Q5 = 1. Roles, circles, and governance processes explicitly decentralize decision rights (Robertson, 2015). Approval flows are minimized.

**SOCIOCRACY 3.0.** Q1–Q5 = 1. Consent-based governance and domain-based circles support local authority and independent action.

**THE FIFTH DISCIPLINE (LEARNING ORGANIZATION).** Q1=0.5, Q2=1, Q3=0.5, Q4=0.5, Q5=0.5. Learning organizations encourage empowerment but do not define explicit authority structures, resulting in mixed support.

**OKR-DRIVEN ORGANIZATIONS.** Q1=0.5, Q2=0, Q3=0, Q4=0, Q5=0.5. OKRs provide local outcome focus, but hierarchically controlled alignment cycles and human-centric approval patterns dominate in practice.

**THEORY OF CONSTRAINTS (TOC).** Q1–Q5 = 1. TOC empowers local teams to elevate, exploit, and manage constraints autonomously (Goldratt and Cox, 1984). Coordination roles are minimal.

### B.6.3 DIMENSION SUMMARY

O2 exposes a strong divide between mechanistic structures (functional, matrix, project-based, capability silos), which centralize authority and inhibit autonomy, and modern decentralized structures (Team Topologies, Platform, Rendanheyi, STS, Teal, Holacracy), which grant robust decision rights to local units. Intermediate models such as Lean Enterprise, Divisional structures, Contingency Theory, and Learning Organizations reveal partial autonomy: their conceptual models support decentralized action, but implementations often retain coordination roles or centralized oversight inherited from legacy structures.

## B.7 DIMENSION O3: STRUCTURAL ADAPTABILITY

This dimension evaluates whether the organizational structure can be reconfigured rapidly and at low cost as new learning emerges. Sociotechnical systems research shows that dynamic, high-uncertainty environments require continuous adjustment of boundaries, roles, and coordination mechanisms (Trist, 1981). Galbraith's information-processing framework likewise predicts that organizations facing high uncertainty must either increase processing capacity or reduce coordination requirements through structural adaptation (Galbraith, 1973). In the centaur ecosystem, where AI systems accelerate discovery and execution, structures that cannot adapt frequently become bottlenecks.

### B.7.1 TERNARY SCORING TABLE FOR O3

Organizational Model	Q1	Q2	Q3	Q4	Q5
Functional Hierarchy	0	0	0	0	0
Matrix Organization	0	0	0	0	0
Project-Based Organization	0	0	0	0	0
Capability-Based (Horizontal Silos)	0	0	0	0	0
Divisional (M-form)	0.5	0.5	0.5	0.5	0.5
Lean Enterprise	1	1	0.5	0.5	1
Spotify Model	1	0.5	1	0.5	0.5
Team Topologies	1	1	1	1	1
Platform Organization Model	1	1	1	1	1
Rendanheyi (Haier)	1	1	1	1	1
Sociotechnical Systems (STS)	1	1	1	1	1
Leavitt's Diamond	0.5	0.5	0.5	0.5	0.5
Galbraith's Information-Processing Model	1	1	0.5	1	1
Contingency Theory	1	0.5	0.5	0.5	0.5
Teal Organizations	1	1	1	1	1
Holacracy	1	1	1	1	1
Sociocracy 3.0	1	1	1	1	1
The Fifth Discipline (Learning Organization)	0.5	0.5	0.5	0.5	0.5
OKR-Driven Organizations	0	0	0.5	0	0
Theory of Constraints (as Org Model)	1	1	1	1	1

**Table 11:** Ternary scoring for O3: Structural Adaptability. Indicators O3-Q1–O3-Q5 are defined in Section B.3.

### B.7.2 RATIONALE FOR EACH ORGANIZATIONAL MODEL

**FUNCTIONAL HIERARCHY.** Q1–Q5 = 0. Structural changes require extensive planning, approval chains, and redefinition of reporting lines. Adaptability is minimal and expensive (Weber, 1978).

**MATRIX ORGANIZATION.** Q1–Q5 = 0. Dual reporting lines make structural change slow and politically negotiated. Adjustments require cross-manager agreement (Galbraith, 1973).

**PROJECT-BASED ORGANIZATION.** Q1–Q5 = 0. Project structures are tied to rigid lifecycles and phase gates (Kerzner, 2017). Structural adaptation mid-stream is rare or discouraged.

**CAPABILITY-BASED SILOS.** Q1–Q5 = 0. Silo boundaries are static; team reconfiguration requires coordination across multiple functional units. Change is slow and high-cost.

**DIVISIONAL (M-FORM).** Q1–Q5 = 0.5. Divisions can adapt portfolios and internal structures with moderate friction (Chandler, 1962). Boundaries evolve, but rarely rapidly or continuously.

**LEAN ENTERPRISE.** Q1=1, Q2=1, Q3=0.5, Q4=0.5, Q5=1. Lean Enterprise strongly supports continuous improvement and rapid local process adaptation (Humble, Molesky, and O'Reilly, 2015). Structural adaptation is partially constrained by underlying legacy patterns (Q3–Q4).

**SPOTIFY MODEL.** Q1=1, Q2=0.5, Q3=1, Q4=0.5, Q5=0.5. Squads adapt easily; chapters/tribes less so. Architectural refactoring is possible but uneven. Spotify explicitly warns that the model “is not static,” leading to mixed scoring.

**TEAM TOPOLOGIES.** Q1–Q5 = 1. The model is expressly designed for continuous, low-cost reconfiguration of team boundaries, interaction modes, and responsibilities (Skelton and Pais, 2019).

**PLATFORM ORGANIZATION MODEL.** Q1–Q5 = 1. Modular platform boundaries allow fluid restructuring of product and platform scopes. Teams and interfaces adapt rapidly with minimal overhead.

**RENDANHEYI (HAIER).** Q1–Q5 = 1. Micro-enterprises dynamically form, dissolve, and re-align around market signals (Wu et al., 2021). Structural adaptability is continuous and intrinsic.

**SOCIOTECHNICAL SYSTEMS (STS).** Q1–Q5 = 1. STS emphasizes iterative redesign of work systems and joint optimization, supporting continuous structural evolution (Trist, 1981).

**LEAVITT'S DIAMOND.** Q1–Q5 = 0.5. The model supports change but does not prescribe how frequently or cheaply it can occur. Adaptability depends on practitioner interpretation.

**GALBRAITH'S INFORMATION-PROCESSING MODEL.** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=1. Galbraith prescribes shifting structures as uncertainty increases (Galbraith, 1973). Some adaptation mechanisms (e.g., teams vs networks) are partially adopted (Q3=0.5).

**CONTINGENCY THEORY.** Q1=1, Q2=0.5, Q3=0.5, Q4=0.5, Q5=0.5. Structural adaptability is theoretically required under uncertainty (Burns and Stalker, 1961), but practical mechanisms are not defined.

**TEAL ORGANIZATIONS.** Q1–Q5 = 1. Self-management and evolutionary purpose support fluid role and boundary updates (Laloux, 2014). Structural change is continuous and local.

**HOLACRACY.** Q1–Q5 = 1. Governance meetings continually redefine roles and accountabilities, enabling frequent structural iteration (Robertson, 2015).

**SOCIOCRACY 3.0.** Q1–Q5 = 1. Circles adapt responsibilities and boundaries through consent-based processes, allowing rapid organizational evolution.

**THE FIFTH DISCIPLINE (LEARNING ORGANIZATION).** Q1–Q5 = 0.5. The concept promotes continuous learning but lacks prescriptive mechanisms for structural reconfiguration (Senge, 1990). Adaptation is conceptual, not structural.

**OKR-DRIVEN ORGANIZATIONS.** Q1=0, Q2=0, Q3=0.5, Q4=0, Q5=0. OKRs support iterative goal-setting but do not alter structure. Org boundaries remain static; adaptations occur annually or ad hoc.

**THEORY OF CONSTRAINTS (TOC).** Q1–Q5 = 1. TOC requires continual experimentation around system constraints (Goldratt and Cox, 1984). Structural refinement is frequent and responsive to evidence.

### B.7.3 DIMENSION SUMMARY

O3 sharply separates rigid, bureaucratic organizational forms (Functional, Matrix, Project-Based, Capability Silos) from modern adaptive structures (Team Topologies, Platform, Rendanheyi, STS, Holacracy, Teal, TOC). Lean Enterprise and Spotify demonstrate moderate adaptability: both promote continuous improvement but are often partially constrained by cultural inertia or residual legacy structures. Analytic models (Leavitt, Contingency, Learning Organization) support adaptability conceptually but lack prescriptive mechanisms for continuous, low-cost structural evolution.

## B.8 DIMENSION O4: OUTCOME ACCOUNTABILITY & VALUE FLOW

This dimension evaluates whether an organizational structure assigns end-to-end accountability for measurable outcomes, integrates quality attributes structurally, and relies on empirical signals to guide decisions. Research in high-performance digital organizations shows that outcome-aligned units, empirical feedback loops, and stable responsibility boundaries correlate strongly with delivery performance and adaptability (Forsgren, Humble, and Kim, 2018). Organizations structured around tasks, deliverables, or functional outputs often fail to provide these conditions.

### B.8.1 TERNARY SCORING TABLE FOR O4

Organizational Model	Q1	Q2	Q3	Q4	Q5
Functional Hierarchy	0	0	0	0	0
Matrix Organization	0	0	0	0	0
Project-Based Organization	0	0	0	0	0
Capability-Based (Horizontal Silos)	0	0	0	0	0
Divisional (M-form)	1	1	1	1	1
Lean Enterprise	0.5	0.5	1	1	1
Spotify Model	1	1	1	1	1
Team Topologies	1	1	1	1	1
Platform Organization Model	1	1	1	1	1
Rendanheyi (Haier)	1	1	1	1	1
Sociotechnical Systems (STS)	1	1	0.5	1	1
Leavitt's Diamond	0.5	0.5	0.5	0.5	0.5
Galbraith's Information-Processing Model	1	1	0.5	1	1
Contingency Theory	1	0.5	0.5	1	0.5
Teal Organizations	0.5	0.5	0.5	0.5	0.5
Holacracy	1	1	1	1	1
Sociocracy 3.0	0.5	0.5	0.5	1	0.5
The Fifth Discipline (Learning Organization)	0.5	0.5	0.5	0.5	0.5
OKR-Driven Organizations	1	1	0.5	1	0.5
Theory of Constraints (as Org Model)	1	1	1	1	1

**Table 12:** Ternary scoring for O4: Outcome Accountability & Value Flow. Indicators O4-Q1–O4-Q5 are defined in Section B.3.

### B.8.2 RATIONALE FOR EACH ORGANIZATIONAL MODEL

**FUNCTIONAL HIERARCHY.** Q1–Q5 = 0. Units are accountable for activities (e.g., development, QA) rather than outcomes. Quality attributes are downstream concerns, and decisions are not guided by empirical signals (Weber, 1978).

**MATRIX ORGANIZATION.** Q1–Q5 = 0. Accountability is fragmented between functional and project managers. No single entity owns end-to-end outcomes or quality attributes (Galbraith, 1973).

**PROJECT-BASED ORGANIZATION.** Q1–Q5 = 0. Projects are defined by deliverables and timelines rather than outcomes (Kerzner, 2017). Quality and value are externalized.

**CAPABILITY-BASED SILOS.** Q1–Q5 = 0. Silos track internal performance metrics; value delivery crosses multiple units with no end-to-end owner.

**DIVISIONAL (M-FORM).** Q1–Q5 = 1. Divisions have clear P&L or market outcomes (Chandler, 1962). Quality attributes and empirical performance signals are embedded in each division's responsibility.

**LEAN ENTERPRISE.** Q1=0.5, Q2=0.5, Q3=1, Q4=1, Q5=1. Lean Enterprise strengthens outcome focus through value-stream alignment and empirical measurement (Humble, Molesky, and O'Reilly, 2015). However, legacy structures often dilute end-to-end accountability (Q1–Q2=0.5).

**SPOTIFY MODEL.** Q1–Q5 = 1. Squads own product outcomes; tribes reinforce domain focus. Quality and metrics are embedded through data-informed iteration.

**TEAM TOPOLOGIES.** Q1–Q5 = 1. Stream-aligned teams hold end-to-end responsibility, with platform and enabling teams supporting quality attributes and empirical feedback (Skelton and Pais, 2019).

**PLATFORM ORGANIZATION MODEL.** Q1–Q5 = 1. Product teams own external outcomes; platform teams own internal service outcomes. Quality attributes and metrics are structured into service contracts.

**RENDANHEYI (HAIER).** Q1–Q5 = 1. Micro-enterprises contractually own customer outcomes (Wu et al., 2021). Execution strategies vary, but outcome responsibility remains stable.

**SOCIOTECHNICAL SYSTEMS (STS).** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=1. STS emphasizes complete socio-technical systems (Trist, 1981). Quality attributes are integrated implicitly but not always explicitly documented (Q3=0.5).

**LEAVITT'S DIAMOND.** Q1–Q5 = 0.5. Outcome orientation is supported conceptually, but not structurally encoded. All indicators receive partial support.

**GALBRAITH'S INFORMATION-PROCESSING MODEL.** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=1. Galbraith emphasizes aligning structure with performance requirements (Galbraith, 1973). Quality attributes may require explicit operationalization (Q3=0.5).

**CONTINGENCY THEORY.** Q1=1, Q2=0.5, Q3=0.5, Q4=1, Q5=0.5. In high-uncertainty contexts, contingency theory supports outcome-driven structures (Burns and Stalker, 1961). Implementation varies widely, leading to partial scores.

**TEAL ORGANIZATIONS.** Q1–Q5 = 0.5. Teal prioritizes purpose and wholeness over formal outcome accountability (Laloux, 2014). Outcome integration and quality attributes are context-dependent.

**HOLACRACY.** Q1–Q5 = 1. Holacracy defines explicit role purposes and domain accountabilities (Robertson, 2015). Quality and empirical signals are integrated through governance.

**SOCIOCRACY 3.0.** Q1=0.5, Q2=0.5, Q3=0.5, Q4=1, Q5=0.5. Outcome accountability exists but is not always structurally stable. Quality and empirical feedback depend on implementation.

**THE FIFTH DISCIPLINE (LEARNING ORGANIZATION).** Q1–Q5 = 0.5. Learning organizations emphasize systems thinking and mental models (Senge, 1990). Structural support for outcome accountability is partial.

**OKR-DRIVEN ORGANIZATIONS.** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=0.5. OKRs define clear, measurable outcomes and empirical key results, but do not inherently integrate quality attributes (Q3=0.5). Outcome stability varies with planning cycles.

**THEORY OF CONSTRAINTS (TOC).** Q1–Q5 = 1. TOC assigns responsibility for system-level throughput, quality, and flow (Goldratt and Cox, 1984). Metrics and constraints anchor outcome stability.

### B.8.3 DIMENSION SUMMARY

O4 highlights a sharp contrast between outcome-oriented structures (Divisional, Lean Enterprise, Team Topologies, Platform, Rendanheyi, Holacracy) and traditional delivery-or-output-based forms (Functional, Matrix, Project, Capability Silos). Conceptual models such as Teal, Sociocracy, the Learning Organization, and Contingency Theory support outcome orientation in principle but lack explicit structural mechanisms. OKR-driven organizations strengthen outcome accountability but do not fully embed quality attributes or value-flow structures. High performers align accountability, feedback, and quality into the organizational fabric rather than relying on upstream analysis or downstream inspection.

## B.9 DIMENSION O5: DISCOVERY INTEGRATION & LEARNABILITY

This dimension evaluates whether discovery—research, experimentation, telemetry analysis, and hypothesis testing—is structurally embedded where value is created. Empirical studies of high-performing digital organizations demonstrate that tight coupling between discovery and delivery accelerates learning and reduces waste (Forsgren, Humble, and Kim, 2018). Lean Thinking (Womack and Jones, 1996), Toyota Kata (Rother, 2010), and Lean Enterprise (Humble, Molesky, and O'Reilly, 2015) similarly emphasize small-batch experimentation, rapid feedback, and continuous improvement. Structures that isolate discovery into upstream units, require centralized approval, or separate analysis from execution impede AI-augmented learning loops.

### B.9.1 TERNARY SCORING TABLE FOR O5

Organizational Model	Q1	Q2	Q3	Q4	Q5
Functional Hierarchy	0	0	0	0	0
Matrix Organization	0	0	0	0	0
Project-Based Organization	0	0	0	0	0
Capability-Based (Horizontal Silos)	0	0	0	0	0
Divisional (M-form)	0.5	1	1	1	1
Lean Enterprise	1	1	1	1	1
Spotify Model	1	1	1	1	1
Team Topologies	1	1	1	1	1
Platform Organization Model	1	1	1	1	1
Rendanheyi (Haier)	1	1	1	1	1
Sociotechnical Systems (STS)	1	1	1	1	1
Leavitt's Diamond	0.5	0.5	0.5	0.5	0.5
Galbraith's Information-Processing Model	1	1	1	1	1
Contingency Theory	0.5	0.5	0.5	0.5	0.5
Teal Organizations	0.5	0.5	0	0.5	0.5
Holacracy	1	1	1	1	1
Sociocracy 3.0	1	1	1	1	1
The Fifth Discipline (Learning Organization)	1	0.5	0.5	0.5	1
OKR-Driven Organizations	0.5	0	0	0	0.5
Theory of Constraints (as Org Model)	1	1	1	1	1

**Table 13:** Ternary scoring for O5: Discovery Integration & Learnability. Indicators O5-Q1–O5-Q5 are defined in Section B.3.

### B.9.2 RATIONALE FOR EACH ORGANIZATIONAL MODEL

**FUNCTIONAL HIERARCHY.** Q1–Q5 = 0. Discovery is isolated upstream (business analysis, research) with long handoffs and no local experiment capability (Weber, 1978).

**MATRIX ORGANIZATION.** Q1–Q5 = 0. Discovery is fragmented; functional and project managers negotiate scope. Experimentation requires cross-managerial approval (Galbraith, 1973).

**PROJECT-BASED ORGANIZATION.** Q1–Q5 = 0. Discovery occurs only in early analysis phases; mid-project learning is costly and discouraged (Kerzner, 2017).

**CAPABILITY-BASED SILOS.** Q1–Q5 = 0. Research and analytics sit in separate departments; delivery teams lack embedded discovery.

**DIVISIONAL (M-FORM).** Q1=0.5, Q2=1, Q3=1, Q4=1, Q5=1. Divisions integrate discovery and delivery but often retain discrete research units. Experimentation autonomy is high, supported by P&L ownership (Chandler, 1962).

**LEAN ENTERPRISE.** Q1–Q5 = 1. Lean Enterprise emphasizes continuous experimentation, small-batch learning, and embedded discovery (Humble, Molesky, and O'Reilly, 2015; Womack and Jones, 1996; Rother, 2010). All indicators are fully met.

**SPOTIFY MODEL.** Q1–Q5 = 1. Squads perform continuous experiments, use data-informed iteration, and integrate research into everyday work.

**TEAM TOPOLOGIES.** Q1–Q5 = 1. Stream-aligned and platform teams support embedded experimentation, telemetry, and automated learning cycles (Skelton and Pais, 2019).

**PLATFORM ORGANIZATION MODEL.** Q1–Q5 = 1. Product teams own experimentation; platform teams provide self-service telemetry, analytics, and experimentation tooling.

**RENDANHEYI (HAIER).** Q1–Q5 = 1. Micro-enterprises continuously experiment based on market signals; discovery is inseparable from execution (Wu et al., 2021).

**SOCIOTECHNICAL SYSTEMS (STS).** Q1–Q5 = 1. STS integrates experimentation within work groups as part of the “joint optimization” loop (Trist, 1981).

**LEAVITT'S DIAMOND.** Q1–Q5 = 0.5. Leavitt emphasizes adjustment of tasks, people, structure, and technology as feedback emerges. Discovery is conceptually supported but not structurally prescribed.

**GALBRAITH'S INFORMATION-PROCESSING MODEL.** Q1–Q5 = 1. Galbraith treats discovery and analysis as information-processing demands; high-learning structures are explicitly supported (Galbraith, 1973).

**CONTINGENCY THEORY.** Q1–Q5 = 0.5. Supports learning conceptually but lacks prescriptive mechanisms for embedded discovery (Burns and Stalker, 1961). Empirical iteration varies.

**TEAL ORGANIZATIONS.** Q1=0.5, Q2=0.5, Q3=0, Q4=0.5, Q5=0.5. Teal encourages learning and reflection (Laloux, 2014) but discovery is inconsistent and often not structurally embedded (Q3=0).

**HOLACRACY.** Q1–Q5 = 1. Governance processes integrate learning into structural adaptation; roles emphasize purpose and continuous improvement (Robertson, 2015).

**SOCIOCRACY 3.0.** Q1–Q5 = 1. Circles embed local experimentation and consent-based learning; decisions incorporate evidence directly.

**THE FIFTH DISCIPLINE (LEARNING ORGANIZATION).** Q1=1, Q2=0.5, Q3=0.5, Q4=0.5, Q5=1. Learning organizations emphasize shared learning (Senge, 1990), but discovery mechanisms are conceptual rather than structurally operationalized.

**OKR-DRIVEN ORGANIZATIONS.** Q1=0.5, Q2=0, Q3=0, Q4=0, Q5=0.5. OKRs clarify outcomes but do not embed discovery. Quarterly cycles limit experimentation autonomy.

**THEORY OF CONSTRAINTS (TOC).** Q1–Q5 = 1. TOC requires continuous experimentation around constraints (Goldratt and Cox, 1984). Embedded learning and empirical decision-making are central to the model.

### B.9.3 DIMENSION SUMMARY

O5 highlights the contrast between structures that deeply embed discovery (Lean Enterprise, Team Topologies, Platform, Rendanheyi, STS, Holacracy, Sociocracy) and those that isolate research upstream or rely on periodic analysis (Functional, Matrix, Project, Silos, OKR). Conceptual organizational models (Leavitt, Contingency Theory, Learning Organization, Teal) encourage learning but lack structural mechanisms for continuous, embedded discovery. High performers tightly couple discovery with execution and support AI-driven insights and automated experimentation.

## B.10 DIMENSION O6: HUMAN–AI COORDINATION COMPATIBILITY

This dimension evaluates whether an organizational structure remains coherent and effective when coordination, monitoring, and analysis increasingly involve AI agents. Research on hybrid human–AI teams shows that agentic systems reduce interpersonal synchronization costs, virtualize shared state, and shift human responsibilities toward intent-setting, oversight, and ethical judgment (Seeber et al., 2020). Distributed cognition theory further emphasizes that cognition extends across humans, tools, and technical agents (Hutchins, 1995). Structures that rely on hierarchical control, human-specific negotiation, or manual task decomposition are poorly suited to such ecosystems. Structures that define coordination abstractly—through responsibilities, interfaces, contracts, or service boundaries—generalize well to agentic contexts.

### B.10.1 TERNARY SCORING TABLE FOR O6

Organizational Model	Q1	Q2	Q3	Q4	Q5
Functional Hierarchy	0	0	0	0	0
Matrix Organization	0	0	0	0	0
Project-Based Organization	0	0	0	0	0
Capability-Based (Horizontal Silos)	0	0	0	0	0
Divisional (M-form)	1	0.5	0.5	1	1
Lean Enterprise	1	1	0.5	1	1
Spotify Model	1	1	0.5	1	1
Team Topologies	1	1	1	1	1
Platform Organization Model	1	1	1	1	1
Rendanheyi (Haier)	1	1	1	1	1
Sociotechnical Systems (STS)	1	1	1	1	1
Leavitt's Diamond	0.5	0.5	0.5	0.5	0.5
Galbraith's Information-Processing Model	1	1	1	1	1
Contingency Theory	1	0.5	0.5	0.5	0.5
Teal Organizations	1	1	0.5	1	0.5
Holacracy	1	1	1	1	1
Sociocracy 3.0	1	1	1	1	1
The Fifth Discipline (Learning Organization)	0.5	1	0.5	1	0.5
OKR-Driven Organizations	0.5	0.5	0	0.5	0
Theory of Constraints (as Org Model)	1	1	1	1	1

**Table 14:** Ternary scoring for O6: Human–AI Coordination Compatibility. Indicators O6-Q1–O6-Q5 are defined in Section B.3.

## B.10.2 RATIONALE FOR EACH ORGANIZATIONAL MODEL

**FUNCTIONAL HIERARCHY.** Q1–Q5 = 0. Coordination depends on hierarchical authority and human supervision (Weber, 1978). Task decomposition and roles assume exclusively human labor.

**MATRIX ORGANIZATION.** Q1–Q5 = 0. Coordination depends on human negotiation between dual reporting lines (Galbraith, 1973). Patterns cannot be expressed as machine-orchestrable interfaces.

**PROJECT-BASED ORGANIZATION.** Q1–Q5 = 0. Phase-gated control, steering committees, and manual sign-offs dominate (Kerzner, 2017). Coordination is human-centric and not agent-generalizable.

**CAPABILITY-BASED SILOS.** Q1–Q5 = 0. Coordination is defined by human specialties and cross-functional negotiation, not interfaces or service boundaries.

**DIVISIONAL (M-FORM).** Q1=1, Q2=0.5, Q3=0.5, Q4=1, Q5=1. Divisional boundaries act as contracts between units (Chandler, 1962). Coordination generalizes well, but roles partially reflect human managerial structures (Q2–Q3=0.5).

**LEAN ENTERPRISE.** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=1. Lean Enterprise removes human bottlenecks via decentralized decision-making and data-driven governance (Humble, Molesky, and O'Reilly, 2015). Some human-specific roles persist in transformation contexts (Q3=0.5).

**SPOTIFY MODEL.** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=1. Tribes and squads coordinate primarily through domains and shared context. Chapters remain partly human-centered (Q3=0.5).

**TEAM TOPOLOGIES.** Q1–Q5 = 1. Coordination occurs through interaction modes and service boundaries, not human hierarchy (Skelton and Pais, 2019). The model explicitly supports agent-mediated interfaces and cognitive-load shaping.

**PLATFORM ORGANIZATION MODEL.** Q1–Q5 = 1. Coordination is expressed via stable interfaces, contracts, and service-level expectations—all machine-orchestrable.

**RENDANHEYI (HAIER).** Q1–Q5 = 1. Micro-enterprises coordinate through digital platform contracts and market signals (Wu et al., 2021). This generalizes directly to agentic coordination.

**SOCIOTECHNICAL SYSTEMS (STS).** Q1–Q5 = 1. STS conceptualizes organizations as distributed cognitive systems, fully compatible with hybrid human–AI coordination (Trist, 1981).

**LEAVITT'S DIAMOND.** Q1–Q5 = 0.5. Coordination is not tied to human psychology, but the model lacks explicit mechanisms for agentic orchestration.

**GALBRAITH'S INFORMATION-PROCESSING MODEL.** Q1–Q5 = 1. Coordination is defined in terms of information-processing requirements, not human roles (Galbraith, 1973). AI augmentation aligns perfectly.

**CONTINGENCY THEORY.** Q1=1, Q2=0.5, Q3=0.5, Q4=0.5, Q5=0.5. Highly generalizable in principle, but coordination patterns remain partially human-specific depending on implementation.

**TEAL ORGANIZATIONS.** Q1=1, Q2=1, Q3=0.5, Q4=1, Q5=0.5. Self-management supports agentic work, but purpose-driven coordination still depends partly on human relational processes.

**HOLACRACY.** Q1–Q5 = 1. Holacracy abstracts roles, accountabilities, and coordination mechanisms into formal, machine-compatible structures (Robertson, 2015).

**SOCIOCRACY 3.0.** Q1–Q5 = 1. Domain-based circle structures and consent processes can be expressed via interfaces; coordination patterns are abstract, not human-specific.

**THE FIFTH DISCIPLINE (LEARNING ORGANIZATION).** Q1=0.5, Q2=1, Q3=0.5, Q4=1, Q5=0.5. Systems thinking aligns well with agentic coordination (Senge, 1990). However, learning processes often rely on human dialogue and reflection.

**OKR-DRIVEN ORGANIZATIONS.** Q1=0.5, Q2=0.5, Q3=0, Q4=0.5, Q5=0. OKRs define outcomes but retain human-centered alignment rituals and quarterly cycles, limiting agentic compatibility.

**THEORY OF CONSTRAINTS (TOC).** Q1–Q5 = 1. TOC defines coordination through system constraints, throughput, and empirical signals (Goldratt and Cox, 1984). All are compatible with automated orchestration.

### B.10.3 DIMENSION SUMMARY

O6 produces the starker contrast across models. Traditional structures (Functional, Matrix, Project-Based, Capability Silos) embed human-centric coordination, managerial negotiation, and hierarchical control, making them poor fits for hybrid human–AI ecosystems. Modern modular structures (Team Topologies, Platform, Rendanheyi, STS, Holacracy, Sociocracy, Lean Enterprise) generalize their coordination logic beyond human roles, enabling agent-mediated orchestration. Conceptual models (Leavitt, Contingency Theory, Learning Organization, Teal) exhibit partial compatibility, with coordination patterns that are abstract enough to permit AI augmentation but not fully structuralized for automated execution.

## B.11 FINAL COMPATIBILITY RESULTS

Using the scoring model defined in Sections B.3–B.4 and the ternary scores assigned in Sections B.5–B.10, each organizational model’s compatibility percentage is computed by averaging its six dimension scores  $O_1 \dots O_6$  and normalizing to a 0–100 scale:

$$\text{Compat\%}(M) = 100 \cdot \frac{\sum_{k=1}^6 \sum_{i=1}^5 q_{k,i}(M)}{30}.$$

Organizational Model	Compatibility (%)
Functional Hierarchy	0%
Matrix Organization	0%
Project-Based Organization	0%
Capability-Based (Horizontal Silos)	0%
OKR-Driven Organizations	33%
The Fifth Discipline (Learning Organization)	58%
Contingency Theory	62%
Leavitt’s Diamond	50%
Teal Organizations	72%
Divisional (M-form)	78%
Lean Enterprise	83%
Sociocracy 3.0	87%
Spotify Model	88%
Theory of Constraints (as Org Model)	95%
Sociotechnical Systems (STS)	97%
Holacracy	97%
Platform Organization Model	98%
Galbraith’s Information-Processing Model	90%
Rendanheyi (Haier)	100%
Team Topologies	100%

**Table 15:** Final compatibility estimates for each organizational model, computed using the ternary-scored organizational dimensions O1–O6.

These results exhibit a structured and theoretically coherent distribution across the twenty organizational models. Traditional mechanistic forms (Functional, Matrix, Project-Based, Capability Silos) demonstrate near-total incompatibility with human–AI software ecosystems, consistent with their reliance on hierarchical control, phase-gated coordination, and rigid functional boundaries. OKR-driven organizations also score low due to their retention of calendar-driven alignment cycles and lack of structural support for continuous discovery or agentic coordination.

Modern product- and platform-oriented structures (Team Topologies, Platform Organizations, Rendanheyi, STS, Holacracy, Spotify) show strong compatibility due to their emphasis on autonomy, outcome ownership, modular boundaries, rapid adaptation, and abstract coordination interfaces that generalize to agent-based orchestration. Lean Enterprise also scores highly by embedding continuous improvement, empirical feedback loops, and value-stream alignment, though legacy structural overlays limit its perfect fit.

Conceptual or analytic models (Leavitt’s Diamond, Contingency Theory, The Fifth Discipline, Teal Organizations) score in the mid-range: their principles align with centaur-compatible dynamics, but they lack prescriptive structural mechanisms such as clear domain boundaries, service contracts, or embedded discovery practices. They offer conceptual guidance but require concrete instantiation to fully support AI-augmented delivery.

Overall, these results reinforce that centaur-compatible organizational design is characterized by autonomy, outcome alignment, structural adaptability, embedded discovery, and abstract coordination patterns—principles shared across Team Topologies, Platform Organizations, Rendanheyi, STS, Holacracy, and Lean Enterprise.

## B.12 METHODOLOGICAL VALIDITY

The methodology used in this appendix extends the structured evaluation approach of Appendix A from delivery frameworks to organizational structures. Instead of assessing process-level assumptions (ritual dependence, coordination mechanisms, iteration models), Appendix B evaluates higher-level sociotechnical architectures along six structural dimensions (O1–O6). These dimensions capture boundary–outcome alignment, autonomy, structural adaptability, outcome accountability, discovery integration, and human–AI coordination compatibility.

The use of a ternary scoring scheme (0, 0.5, 1) is consistent with empirical practice in organizational research, where graded assessments are preferred to binary judgments due to the inherent partiality, context-dependence, and heterogeneity of organizational forms (Mintzberg, 1979; Galbraith, 1973; Burns and Stalker, 1961; Trist, 1981). Each indicator corresponds to a theoretically grounded property drawn from sociotechnical systems theory, contingency theory, and information-processing perspectives, as summarized in Section B.2.

The method belongs to a class of scientifically recognized techniques for evaluating complex sociotechnical systems under uncertainty:

- **Structured Expert Judgment (SEJ)** (Cooke, 1991) decomposes complex constructs into smaller, theoretically motivated indicators, then evaluates each transparently.
- **Multi-criteria assessment** is widely used in organizational and systems design to compare alternatives across heterogeneous criteria, often using ordinal or ternary scales.
- **Sociotechnical systems analysis** explicitly considers alignment between organizational structures and technical systems (Trist, 1981; Sailer and McCulloh, 2020).

The organizational compatibility model is scientifically valid in this context because:

- All dimensions (O1–O6) are derived from established theories of organizational fit, coordination, and sociotechnical alignment.
- Indicators are explicit and interpretable; each ternary score is traceable to a specific property.
- The aggregation function (simple mean across indicators and dimensions) is monotonic, transparent, and does not introduce hidden weighting.
- Ternary values (0, 0.5, 1) allow the model to distinguish between absence, partial presence, and strong presence of an organizational affordance.

The model is not intended to predict performance directly, but to assess structural compatibility of organizational designs with the demands of human–AI software development ecosystems. Its value lies in making implicit assumptions explicit and open to critique, replication, and refinement.

## B.13 LIMITATIONS

Despite its rigor, the organizational compatibility model presented in this appendix has several limitations.

**CONCEPTUAL RATHER THAN EMPIRICAL VALIDATION.** As with Appendix A, there is limited empirical data on fully agentic human–AI development ecosystems. The compatibility scores assess structural fit based on current theory, not measured performance outcomes in future-state organizations.

**ABSTRACTNESS OF SOME MODELS.** Several organizational models (e.g., Leavitt’s Diamond, Contingency Theory, The Fifth Discipline) are conceptual frameworks rather than prescriptive blueprints. Their practical impact depends heavily on concrete instantiation. The ternary scoring system accommodates this via partial scores (0.5), but interpretation is still required.

**IMPLEMENTATION VARIANCE.** Organizational forms such as Lean Enterprise, Spotify, Teal, Holacracy, or Sociocracy 3.0 exhibit wide variation in practice. An organization may claim to adopt a model while preserving legacy practices that substantially alter its actual properties. This evaluation scores canonical definitions, not specific implementations.

**INDICATOR INDEPENDENCE ASSUMPTION.** Dimensions O1–O6 are treated as independent for scoring purposes, although in reality they may be correlated (e.g., autonomy often co-occurs with better discovery integration). Nonlinear interaction effects are not modeled explicitly.

**STRUCTURED EXPERT JUDGMENT.** As with any structured expert judgment method (Cooke, 1991; Aspinall, 2010), some subjectivity remains in interpreting how well a given organizational model satisfies each indicator. Ternary scoring reduces ambiguity but does not eliminate it entirely.

These limitations do not invalidate the analysis but should be considered when interpreting the scores. The framework is best used as a comparative and diagnostic tool, not as a deterministic predictor of organizational success.

## B.14 REPRODUCIBILITY AND INTER-RATER RELIABILITY

Reproducibility is supported by the explicit definition of all dimensions (O1–O6) and indicators, as well as the complete scoring tables and rationales provided in Sections B.5–B.10. Any researcher can independently rescore the twenty organizational models using:

1. the same set of dimensions and indicator definitions,
2. the same canonical descriptions of the organizational models,
3. the same ternary scoring scale and aggregation formula.

Inter-rater reliability can be improved through:

- calibration sessions using a subset of organizational models to align interpretations of indicators,
- independent scoring by multiple experts followed by discussion of discrepancies,
- documenting explicit scoring criteria and examples for each indicator,
- sensitivity analysis to evaluate the impact of alternative interpretations or scoring thresholds.

These practices mirror those recommended in structured expert judgment (Cooke, 1991; Aspinall, 2010) and sociotechnical systems analysis (Trist, 1981). While some variance between raters is expected, the explicit indicator set and transparent scoring tables create a robust foundation for reproducible comparative work.

## B.15 CONCLUSION

Appendix B extends the centaur-compatibility analysis of Appendix A from delivery frameworks to organizational structures. By evaluating twenty organizational models against six structural dimensions—boundary–outcome alignment, autonomy, structural adaptability, outcome accountability, discovery integration, and human–AI coordination compatibility—the appendix makes explicit which organizational forms are structurally well-suited to human–AI software development ecosystems and which are not.

The results show that traditional mechanistic structures (Functional Hierarchy, Matrix Organization, Project-Based Organizations, Capability-Based Silos) are fundamentally misaligned with centaur-era demands. Their reliance on hierarchical control, human-centric coordination, rigid boundaries, and phase-gated discovery contradicts the core properties of AI-augmented development: cheap coordination, continuous experimentation, and outcome-focused flow.

In contrast, modern product and platform structures (Divisional models, Lean Enterprise, Spotify-style product organizations, Team Topologies, Platform Organizations, Rendanheyi) and sociotechnical or post-bureaucratic models (STS, Holacracy, Sociocracy 3.0, Teal, TOC) exhibit much higher compatibility. They emphasize outcome ownership, autonomy, modular boundaries, continuous learning, and coordination via interfaces or contracts—all favorable conditions for integrating AI agents into development workflows.

This appendix underscores that organizational form is as critical as delivery method in determining whether human–AI development can realize its potential. A centaur-compatible delivery framework such as OBAF will struggle inside a structurally hostile organization, while a structurally supportive organization enables frameworks like OBAF, Lean Startup, or DevOps/SRE to amplify their impact. Together, Appendices A and B provide a multi-level compatibility map for navigating the transition toward AI-augmented, outcome-based software engineering.

## APPENDIX C: ORGANIZATIONAL ALIGNMENT WITH OBAF

### C.1 INTRODUCTION

Appendices A and B evaluated delivery frameworks and organizational structures, respectively, against the structural and epistemic requirements of human–AI (“centaur”) software development ecosystems. These analyses assessed how well each framework or organizational design generalizes to a future characterized by autonomous coding agents, continuous discovery, cheap iteration, and hybrid human–AI cognitive loops.

However, centaur compatibility alone does not determine whether an organizational structure will support specific delivery methods. A structure may be well suited to the emerging human–AI reality and still be poorly aligned with a particular framework’s principles, assumptions, or epistemic requirements. For example, several sociotechnical and post-bureaucratic structures (e.g., Teal, Holacracy, Sociocracy 3.0) scored highly in Appendix B due to their adaptability, autonomy, and abstract coordination models—yet these same features may introduce friction or misalignment when paired with a delivery method that depends on clear outcome ownership, evidence-driven iteration, and simple enabling governance.

To make this distinction explicit, this appendix evaluates how well each of the twenty organizational structures from Appendix B aligns with the tenets of the Outcome-Based Agile Framework (OBAF) (Blomgren, 2025b). OBAF emphasizes:

- outcomes over requirements or outputs,
- constraints over scope,
- problem ownership over task execution,
- continuous discovery integrated with delivery,
- hypotheses over plans (evidence as the arbiter),
- leadership through intent and constraints (not instruction),
- simple, enabling governance,
- blameless learning through After Action Reviews (AARs),
- evidence-framing conversations rather than status reporting.

OBAF is not a ritualized framework but an epistemic model of adaptive development; its success depends on organizational conditions that enable outcome definition, problem reframing, continuous discovery, and evidence-driven decision-making. An organizational structure incapable of supporting these conditions will generate structural friction, independent of its compatibility with the centaur development paradigm.

To evaluate this alignment, we introduce a new organizational dimension—denoted  $F_1$ —which assesses the extent to which an organizational structure supports or obstructs the conditions required for OBAF to function effectively. As with the organizational dimen-

sions in Appendix B, we use a ternary scoring system (0, 0.5, 1) to capture full, partial, or absent alignment.

### C.2 METHODOLOGY: OBAF ORGANIZATIONAL ALIGNMENT ( $F_1$ )

OBAF’s tenets establish a set of structural and cultural preconditions that must be supported by the organization for the framework to function. These include: durable outcome ownership, local autonomy to reformulate problems, integration of discovery and delivery, evidence-framed conversations, leadership by intent and constraints, lightweight enabling governance, and a culture of blameless learning.

The  $F_1$  dimension evaluates these preconditions through eight indicators, each scored using a ternary scale:

$$q_i(M) \in \{0, 0.5, 1\}.$$

A score of:

- 1 indicates strong structural and cultural support for the indicator;
- 0.5 indicates partial, conditional, or uneven support (e.g., possible but requiring adaptation);
- 0 indicates structural misalignment or contradiction.

### C.2.1 INDICATORS FOR OBAF ORGANIZATIONAL ALIGNMENT

Each organizational model  $M$  is evaluated against the following eight indicators, derived from OBAF's core tenets (Blomgren, 2025b) and supported by empirical findings in Lean, DevOps, and learning organizations (Forsgren, Humble, and Kim, 2018; Womack and Jones, 1996; Rother, 2010; Senge, 1990):

1. **F1-Q1: Outcome-based structuring.** Does the structure make outcome ownership natural and durable? (0 = output/task-based; 0.5 = mixed; 1 = outcome-based)
2. **F1-Q2: Problem ownership vs task execution.** Do teams own problems within constraints, rather than executing predefined tasks? (0 = task-driven; 0.5 = partial; 1 = problem-driven)
3. **F1-Q3: Continuous discovery integration.** Does the structure support ongoing experimentation and discovery at the team level? (0 = centralized or phase-gated discovery; 0.5 = partially integrated; 1 = fully integrated)
4. **F1-Q4: Evidence-framed decision processes.** Are decisions framed around empirical evidence rather than status updates or plan conformance? (0 = status/plan-driven; 0.5 = mixed; 1 = evidence-driven)
5. **F1-Q5: Leadership by intent and constraints.** Does leadership provide direction via intent, constraints, and context rather than instruction? (0 = command/control; 0.5 = mixed; 1 = intent-driven)
6. **F1-Q6: Governance simplicity & enablement.** Is governance light, enabling, and designed to remove friction rather than enforce compliance? (0 = heavy/compliance; 0.5 = mixed; 1 = enabling)
7. **F1-Q7: Blameless learning culture (AAR compatibility).** Does the structure allow blameless reflection and continuous learning via AARs? (0 = blame/inspection; 0.5 = mixed; 1 = blameless learning)
8. **F1-Q8: Evidence-framing instead of status-reporting.** Are progress discussions anchored in evidence, insights, and next hypotheses? (0 = status/activity focus; 0.5 = mixed; 1 = evidence-framing)

### C.2.2 SCORING FORMULA

The OBAF-alignment score for organizational model  $M$  is computed as:

$$F_1(M) = \frac{1}{8} \sum_{i=1}^8 q_i(M),$$

with a normalized percentage:

$$\text{OBAF-Align\%}(M) = 100 \cdot F_1(M).$$

High scores indicate strong structural and cultural support for OBAF's tenets, enabling the framework to operate with minimal friction. Low scores indicate structural contradictions, suggesting that OBAF would require extensive adaptation—or may not be viable—within that organizational form.

Sections C.3–C.4 present the full scoring tables, rationales, and comparative analysis of OBAF alignment across the twenty organizational models evaluated in Appendix B.

### C.3 F1 SCORING TABLE: ORGANIZATIONAL ALIGNMENT WITH OBAF TENETS

Organizational Model	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Functional Hierarchy	0	0	0	0	0	0	0	0
Matrix Organization	0	0	0	0	0	0	0	0
Project-Based Organization	0	0	0	0	0	0	0	0
Capability-Based (Horizontal Silos)	0	0	0	0	0	0	0	0
Divisional (M-form)	1	0.5	0.5	1	0.5	0.5	0.5	1
Lean Enterprise	1	1	1	1	1	1	1	1
Spotify Model	1	1	1	1	0.5	0.5	1	1
Team Topologies	1	1	1	1	1	1	1	1
Platform Organization Model	1	1	1	1	1	1	1	1
Rendanheyi (Haier)	1	1	1	1	1	1	1	1
Sociotechnical Systems (STS)	1	1	1	1	1	1	1	1
Leavitt's Diamond	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Galbraith's Information-Processing Model	1	0.5	1	1	0.5	0.5	1	1
Contingency Theory	0.5	0.5	0.5	1	0.5	0.5	0.5	0.5
Teal Organizations	0.5	1	0.5	0.5	1	1	1	0.5
Holacracy	1	1	1	1	1	1	1	1
Sociocracy 3.0	1	1	1	1	1	1	1	1
The Fifth Discipline (Learning Organization)	0.5	1	0.5	1	0.5	0.5	1	0.5
OKR-Driven Organizations	0.5	0.5	0	0.5	0.5	0	0.5	0
Theory of Constraints (as Org Model)	1	1	1	1	1	1	1	1

**Table 16:** Ternary scoring for F1: OBAF Organizational Alignment. Indicators F1-Q1–F1-Q8 are defined in Section C.2.

### C.4 RATIONALE FOR EACH ORGANIZATIONAL MODEL

This section provides rationale for the OBAF-alignment scores in Table C.3, explaining how each organizational model supports or obstructs the core OBAF tenets: outcome orientation, problem ownership, continuous discovery, evidence framing, intent-based leadership, enabling governance, blameless learning, and evidence-driven progress conversations.

**FUNCTIONAL HIERARCHY.** Functional hierarchies structure work around internal activities rather than outcomes. Task execution is prescribed by upstream planning; discovery is centralized or phase-gated; decisions are status- and plan-driven; leadership is command-oriented; governance is compliance-heavy; and learning mechanisms (if present) tend to be inspection- or blame-oriented (Weber, 1978). This structure contradicts every OBAF tenet.

**MATRIX ORGANIZATION.** Matrix organizations fragment accountability across dual reporting lines (Galbraith, 1973). Teams rarely own problems; discovery requires cross-manager approval; evidence is filtered through managerial layers; governance is complex and coordination-heavy; and performance conversations emphasize status and alignment rituals. The structure is fundamentally incompatible with OBAF’s emphasis on autonomy, evidence, and outcome ownership.

**PROJECT-BASED ORGANIZATION.** Projects emphasize deliverables, milestones, and status reporting (Kerzner, 2017). Problem ownership is temporary and dissolves at project end; discovery is phase-gated; governance is oriented toward plan adherence; leadership sets solutions rather than intent; and learning is typically retrospective in a blame-sensitive context. OBAF’s tenets conflict directly with project-mode work.

**CAPABILITY-BASED SILOS.** Silos organize around specialist skills rather than outcome responsibility. Teams execute assigned tasks; discovery is upstream; evidence-framed decision-making is rare; governance is heavy; leadership is solution-driven; and AAR-like learning is fragmented across departmental boundaries. The structure obstructs all OBAF tenets.

**DIVISIONAL (M-FORM).** Divisions own outcomes and markets (Chandler, 1962), supporting OBAF's outcome-driven tenets (F1-Q1, Q4). However, legacy coordination roles weaken problem ownership (0.5), discovery integration (0.5), and enabling governance (0.5). Divisions often retain status-reporting behaviors inherited from hierarchical structures (0.5). Overall support is moderate to strong but not complete.

**LEAN ENTERPRISE.** Lean Enterprise promotes continuous learning, value-stream orientation, empowered teams, evidence-driven decisions, intent-based leadership, and blameless improvement cycles (Humble, Molesky, and O'Reilly, 2015; Womack and Jones, 1996; Rother, 2010). All OBAF tenets are structurally reinforced: problem ownership, discovery integration, hypothesis-driven planning, lightweight governance, and evidence framing. It earns full alignment across all indicators.

**SPOTIFY MODEL.** Spotify squads own problems and outcomes; tribes provide intent; discovery is embedded; evidence-driven iteration is encouraged; and learning is culturally central. However, chapters/tribes introduce coordination roles that may complicate governance simplicity (0.5) and leadership intent purity (0.5). Status alignment rituals still appear in practice. High but not perfect alignment.

**TEAM TOPOLOGIES.** Team Topologies places outcome ownership, problem framing, enabling governance, evidence-driven change, and continuous learning at its core (Skelton and Pais, 2019). All interaction modes reinforce evidence framing, and AAR-style learning is structurally embedded. It is one of the strongest matches to OBAF's tenets, earning full alignment.

**PLATFORM ORGANIZATION MODEL.** Platform organizations emphasize product outcomes, problem-focused autonomy, embedded discovery, evidence-driven decisions, service-based governance, and continuous learning. Governance is lightweight and enabling; leadership is intent-driven; progress is defined by evidence, not status. Strong OBAF alignment across all indicators.

**RENDANHEYI (HAIER).** Micro-enterprises own outcomes, redefine problems locally, run continuous experiments, and coordinate via digital contracts (Wu et al., 2021). Leadership provides intent and constraints through ecosystem governance. Learning is blameless and continuous. Status reporting is minimal; evidence governs decisions. Near-perfect OBAF alignment.

**SOCIOTECHNICAL SYSTEMS (STS).** STS emphasizes joint optimization, autonomous problem-solving units, continuous feedback, and learning (Trist, 1981). STS teams own problems, integrate discovery, and rely on empirical signals. Governance is minimal and enabling. Strong alignment with OBAF, although explicit evidence-framing mechanisms vary across implementations.

**LEAVITT'S DIAMOND.** Leavitt's model encourages adaptation of tasks, people, structure, and technology but does not prescribe explicit mechanisms for outcome ownership, continuous discovery, evidence framing, or governance style. Each OBAF-aligned property is conceptually supported but incompletely expressed, resulting in partial alignment across all indicators.

**GALBRAITH'S INFORMATION-PROCESSING MODEL.** Galbraith emphasizes empirical information flows, uncertainty reduction, and structural adaptation (Galbraith, 1973). Problem ownership and discovery integration are conceptually supported; evidence framing is strongly aligned. However, leadership and governance simplicity depend heavily on implementation, giving mixed scores on those indicators.

**CONTINGENCY THEORY.** Contingency theory argues that structure must fit environmental demands (Burns and Stalker, 1961). Outcome ownership and evidence framing are conceptually supported, but the theory is silent on governance simplicity, blameless learning, and discovery integration, yielding mixed alignment.

**TEAL ORGANIZATIONS.** Teal self-management supports problem ownership and autonomy (Laloux, 2014). Purpose-driven alignment fits OBAF's outcome focus. However, discovery practices, evidence framing, and governance simplicity vary widely; Teal's reliance on relational trust sometimes inhibits structured AARs or hypothesis-driven decision-making. Mixed alignment.

**HOLACRACY.** Holacracy's explicit role definitions, integrative decision-making, tension processing, and governance mechanisms map closely to OBAF's principles (Robertson, 2015). Roles own problems and outcomes; discovery feeds governance; evidence frames decisions; leadership is intent-based; and learning is continuous. Strong alignment across all indicators.

**SOCIOCRACY 3.0.** Consent-based decision-making, domain-based circles, and evidence-guided proposals strongly support OBAF. Problem ownership, embedded discovery, blameless learning, and enabling governance are built into its practices. Structural alignment with OBAF is nearly complete.

**THE FIFTH DISCIPLINE (LEARNING ORGANIZATION).** Learning Organizations support systems thinking, reflection, evidence awareness, and psychological safety (Senge, 1990). These enable intent-driven leadership and blameless learning. However, problem ownership, discovery integration, and governance simplicity depend on implementation, leading to partial alignment.

**OKR-DRIVEN ORGANIZATIONS.** OKRs strengthen outcome focus and evidence-based progress tracking but rarely embed continuous discovery or intent-based leadership. Leadership alignment cycles can be hierarchical; governance is often compliance-driven; and discovery is not structurally integrated. Mixed-to-low alignment across OBAF tenets.

**THEORY OF CONSTRAINTS (TOC).** TOC centers on empirical evidence, continuous discovery around constraints, autonomous problem ownership, and system-level learning (Goldratt and Cox, 1984). Governance is lightweight; leadership provides intent; and progress is evidence-framed. TOC aligns exceptionally well with OBAF, matching all indicators.

## C.5 FINAL OBAF ALIGNMENT RESULTS

Using the scoring model defined in Section C.2 and the ternary scores presented in Section C.3, each organizational model’s OBAF-alignment percentage is computed as:

$$\text{OBAF-Align\%}(M) = 100 \cdot \frac{1}{8} \sum_{i=1}^8 q_i(M).$$

Organizational Model	OBAF Alignment (%)
Functional Hierarchy	0%
Matrix Organization	0%
Project-Based Organization	0%
Capability-Based (Horizontal Silos)	0%
OKR-Driven Organizations	19%
Leavitt’s Diamond	50%
The Fifth Discipline (Learning Organization)	56%
Contingency Theory	56%
Teal Organizations	69%
Divisional (M-form)	69%
Lean Enterprise	100%
Spotify Model	88%
Galbraith’s Information-Processing Model	81%
Sociocracy 3.0	100%
Holacracy	100%
Team Topologies	100%
Platform Organization Model	100%
Rendanheyi (Haier)	100%
Sociotechnical Systems (STS)	100%
Theory of Constraints (as Org Model)	100%

**Table 17:** Final OBAF-alignment percentages for twenty organizational models, computed from F1–Q1–F1–Q8.

The OBAF-alignment results reveal three clusters of organizational structures. The first cluster (Functional, Matrix, Project-Based, Capability Silos) exhibits complete misalignment with OBAF’s tenets: these structures emphasize task execution, hierarchical control, status-driven governance, and centralized decision-making, all of which contradict OBAF’s outcome-driven, evidence-based, and discovery-integrated philosophy.

A large middle cluster (Leavitt’s Diamond, Contingency Theory, The Fifth Discipline, Teal Organizations, Divisional structures, Galbraith’s information-processing model, Spotify) shows partial alignment. These structures can support problem ownership, evidence framing, and continuous learning, but often lack strong mechanisms for outcome ownership, simple enabling governance, or deeply embedded discovery. They can host OBAF but usually require structural or cultural adaptation.

The final cluster (Lean Enterprise, Sociocracy 3.0, Holacracy, Team Topologies, Platform Organizations, Rendanheyi, STS, and TOC) demonstrates strong alignment across almost all OBAF indicators. These models naturally emphasize outcome ownership, autonomy, continuous discovery, evidence-based adaptation, governance minimalism, and blameless learning—structural conditions that closely match OBAF’s epistemic foundation.

This result highlights that even though several organizational models are highly compatible with the centaur reality (see Appendix B), a smaller subset is truly optimized for OBAF’s tenets, confirming that centaur-compatibility and OBAF-compatibility are related but distinct organizational properties.

## C.6 DISCUSSION: ORGANIZATIONAL FIT FOR OBAF

The OBAF-alignment results in Section C.5 allow us to distinguish between organizational structures that naturally support OBAF’s epistemic foundations and those that would require significant structural or cultural transformation. Importantly, OBAF compatibility is not equivalent to centaur compatibility. Several organizational models that scored highly in Appendix B (e.g., Teal, STS, Sociocracy) exhibit strong centaur properties—such as autonomy, adaptability, and abstract coordination—yet differ in their support for OBAF’s specific tenets concerning outcomes, evidence framing, simple enabling governance, and structured discovery integration.

This section synthesizes the results into three categories: organizational models that are natural hosts for OBAF, models that can support OBAF with adaptation, and models that are structurally hostile to OBAF due to deep misalignments with its principles.

**C.6.1 NATURAL HOSTS FOR OBAF.** Structures scoring 100% alignment (Lean Enterprise, Sociocracy 3.0, Holacracy, Team Topologies, Platform Organization Model, Rendanheyi, Sociotechnical Systems, Theory of Constraints) exhibit strong structural support for all OBAF indicators. These models share several characteristics that closely mirror OBAF’s tenets:

- **Outcome-centric boundaries** that make outcome ownership persistent and visible.
- **High autonomy and problem ownership**, enabling teams to frame and reframe problems within constraints.
- **Embedded continuous discovery**, ensuring that learning and experimentation are integrated with delivery.
- **Evidence-driven decision processes** and hypothesis-based planning.
- **Leadership expressed as intent and constraints**, rather than prescriptive task direction.
- **Simple, enabling governance** with minimal ceremony or compliance-based oversight.
- **Cultural reinforcement of blameless learning**, closely aligned with OBAF’s emphasis on After Action Reviews.

These organizational structures provide nearly all preconditions for OBAF to function without modification. In practice, organizations implementing OBAF within these designs face minimal structural resistance. Team Topologies and Platform Organization Models stand out as particularly strong complements: their boundary logic, interaction modes, and cognitive-load principles dovetail directly with OBAF’s emphasis on outcomes, constraints, and continuous discovery.

## C.6.2 CONDITIONAL OR ADAPTABLE HOSTS.

A substantial set of organizational structures—Divisional, Spotify, Galbraith’s information-processing view, Teal, Contingency Theory, and The Fifth Discipline—achieve moderate-to-high OBAF alignment (50–88%). These models provide partial or uneven support for OBAF, often due to structural or cultural ambiguities:

- **Divisional structures** strongly support outcomes but frequently retain legacy coordination roles, diluted problem ownership, and hierarchical reporting mechanisms that hinder evidence framing.
- **Spotify’s model** embeds discovery and outcome ownership but introduces cross-cutting roles (chapters, tribes) that complicate governance simplicity and leadership intent.
- **Galbraith’s model** emphasizes empirical information processing but lacks explicit guidance for governance simplicity, problem ownership, or blameless learning.
- **Teal organizations** emphasize self-management and purpose but offer weak structural support for evidence framing, discovery integration, and governance minimalism.
- **Contingency Theory and The Fifth Discipline** support OBAF conceptually but do not prescribe structural mechanisms for discovery integration, lightweight governance, or steady outcome ownership.

These structures can host OBAF, but doing so requires targeted adaptations—often involving clearer domain boundaries, explicit embedding of discovery practices, or the introduction of formalized evidence-framing mechanisms. Without such adaptations, teams may struggle to consistently enact core OBAF habits such as hypothesis-driven planning, recognitional problem framing, or continuous discovery.

### C.6.3 STRUCTURALLY HOSTILE CONTEXTS.

Functional hierarchies, matrix organizations, project-based organizations, and capability-based silos score 0% alignment. Their misalignment with OBAF arises from deep structural and cultural assumptions:

- **Outcome ownership is absent** due to task- or deliverable-based boundaries.
- **Discovery is centralized or phase-gated**, contradicting OBAF’s integrated discovery loop.
- **Evidence framing is replaced by status reporting** and plan-based oversight.
- **Leadership is command-oriented** and solution-prescriptive rather than intent-driven.
- **Governance is heavy, ritualistic, and compliance-centered**, opposite OBAF’s enabling governance principles.
- **Learning is blame- or inspection-oriented**, conflicting with OBAF’s AAR philosophy.

No amount of OBAF coaching or team-level adoption can compensate for these structural contradictions. Organizations wishing to apply OBAF in such environments must perform substantial structural reform—typically moving toward value-stream alignment, reducing hierarchical control, and replacing reporting cycles with evidence-framed governance.

### C.6.4 RELATIONSHIP BETWEEN CENTAUR COMPATIBILITY AND OBAF COMPATIBILITY.

Appendix B demonstrated that many organizational structures—particularly Lean Enterprise, STS, Platform Organizations, Rendanheyi, and Team Topologies—are highly compatible with the centaur development paradigm. This appendix reveals that many of these same structures are also highly compatible with OBAF. This is not coincidental: OBAF’s tenets (outcomes, evidence, continuous discovery, recognitional problem framing, and constraint-based planning) are epistemically aligned with the demands of human–AI development ecosystems.

However, several models (Teal, Contingency Theory, Fifth Discipline) scored high on centaur compatibility but only moderately on OBAF alignment. These structures support autonomy and adaptability but do not provide the structural mechanisms required for OBAF’s consistent evidence framing, governance simplicity, or steady outcome ownership. Conversely, TOC and Holacracy scored exceptionally high on both analyses, suggesting that they represent structurally and epistemically robust foundations for implementing OBAF at scale.

### C.6.5 IMPLICATIONS FOR ORGANIZATIONAL DESIGN.

The results highlight two independent but related strategic questions for organizations:

1. **Is the organizational structure compatible with the centaur development reality?** (Appendix B)
2. **Is the organizational structure compatible with the chosen delivery method?** (Appendix C)

A structure may support one but not the other. The most effective configurations are those that simultaneously:

- align boundaries to outcomes and domains,
- distribute decision-making to autonomous units,
- integrate discovery tightly with delivery,
- minimize hierarchical or ritualistic governance,
- enable blameless and evidence-framed learning,
- and provide interfaces that generalize to human–AI coordination.

In practice, organizations seeking to adopt OBAF should either:

- choose structures that already embody these conditions (e.g., Team Topologies, Platform Organizations, Lean Enterprise, Rendanheyi, STS, TOC, Holacracy, Sociocracy 3.0), or
- transform their existing structures to incorporate these properties deliberately.

When these organizational conditions are met, OBAF’s tenets do not merely “fit”—they amplify the organization’s ability to deliver adaptive, outcome-driven value in hybrid human–AI environments.

## C.7 LIMITATIONS

While the OBAF-alignment model introduced in this appendix provides a structured and transparent evaluation of organizational compatibility, several limitations must be acknowledged.

**CONCEPTUAL ABSTRACTION.** The OBAF-alignment dimension (F1) evaluates organizational structures at a conceptual level. Many models—such as Leavitt’s Diamond, Contingency Theory, and The Fifth Discipline—are intentionally abstract and require concrete instantiation before their practical impact on OBAF can be observed. The ternary scoring approach mitigates this by allowing partial alignment, but conceptual interpretation remains necessary.

## VARIATION IN ORGANIZATIONAL IMPLEMENTATION.

Organizational models such as Lean Enterprise, Spotify, Holacracy, and Sociocracy 3.0 exhibit wide variation in real-world implementation. A structure that aligns strongly with OBAF in theory may be implemented in a way that undermines OBAF's tenets due to cultural inertia, leadership styles, or historical constraints. This model scores structures based on their canonical definitions rather than their common misapplications.

## DEPENDENCE ON INTERPRETIVE JUDGMENT.

Although the ternary scoring system reduces ambiguity relative to binary scoring, some degree of expert judgment is unavoidable. OBAF's tenets—such as evidence framing or intent-based leadership—can be partially satisfied in multiple ways, and different interpretive perspectives may yield modest variations in scores. However, complete indicator transparency ensures replicability and opens the evaluation to critique.

## OBAF-SPECIFIC BIAS BY DESIGN.

This appendix explicitly evaluates organizational alignment with OBAF's tenets, not with alternative delivery frameworks or organizational theories. As such, structures optimized for different philosophies (e.g., Teal's evolutionary purpose or Sociocracy's consent-based governance) may score lower on OBAF alignment despite performing well in other contexts or even excelling in the centaur compatibility analysis of Appendix B.

**FUTURE-STATE UNCERTAINTY.** As with all analyses in this document, the lack of empirical data from fully mature human–AI development ecosystems introduces inherent uncertainty. Organizational structures may interact with AI-enabled workflows in ways not yet fully understood. The model evaluates structural compatibility based on current research; empirical insight from future agentic ecosystems may require refinement of the indicators.

## INTERACTION EFFECTS NOT MODELED.

The OBAF-alignment model evaluates each indicator independently, though real-world organizational effectiveness depends on complex interactions between boundaries, leadership, incentives, culture, and technical architecture. Nonlinear or emergent effects are therefore not captured directly in the scoring model.

Despite these limitations, the framework remains useful for comparative analysis, organizational self-assessment, structural critique, and strategic alignment planning. Its transparent criteria and ternary scoring mitigate bias and promote replicability.

## C.8 CONCLUSION

This appendix introduced a structured methodology for evaluating how well twenty organizational models align with the tenets of the Outcome-Based Agile Framework (OBAF). The analysis complements the centaur compatibility results in Appendix B by assessing a different but equally important question: given the structural and cultural assumptions embedded within each organizational form, which models naturally support OBAF's outcome-driven, evidence-centered, discovery-integrated philosophy?

The results demonstrate that OBAF alignment is not a byproduct of general adaptability or autonomy. Several organizational structures—such as Teal, Contingency Theory, and The Fifth Discipline—show strong compatibility with the emerging human–AI future yet only partial alignment with OBAF. Conversely, models such as Team Topologies, Platform Organizations, STS, Sociocracy 3.0, Holacracy, Lean Enterprise, Rendanheyi, and Theory of Constraints exhibit near-complete alignment across all indicators. These structures encode many of the conditions OBAF requires: durable outcome ownership, problem framing within constraints, embedded discovery, evidence-based decision-making, lightweight enabling governance, and blameless learning cultures.

At the opposite extreme, traditional models—Functional, Matrix, Project-Based, and Capability-Based Silos—remain fundamentally incompatible with OBAF. Their emphasis on hierarchical control, task decomposition, phase-gated discovery, status reporting, and rigid boundaries contradict OBAF's foundational assumptions. Team-level adoption of OBAF is unlikely to succeed within such structures without substantial organizational transformation.

Taken together, Appendices A, B, and C provide a multi-layered view of compatibility between delivery frameworks, organizational structures, and the emerging human–AI development paradigm. OBAF's strong alignment with the centaur reality—and its compatibility with several high-performing modern organizational models—suggests that outcome-based, evidence-driven, continuously adaptive delivery is not only feasible but increasingly necessary as agentic systems reshape the cognitive and structural landscape of software development.

This appendix provides organizations with an evaluative tool for determining whether their structural foundations support or obstruct OBAF's principles, enabling informed strategic decisions about organizational transformation, framework adoption, or hybrid approaches tailored to the demands of centaur-era software engineering.

## REFERENCES

- Alves, Pedro and Bruno Pereira Cipriano (2023). “The Centaur Programmer – How Kasparov’s Advanced Chess spans over to the software development of the future”. In: arXiv preprint arXiv:2304.11172. Preprint; CC BY 4.0. url: <https://arxiv.org/abs/2304.11172>.
- Anderson, David J (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.
- Aspinall, Willy (2010). “Scientists and Structured Expert Judgment”. In: Environment and Development.
- Bavishi, Rohan et al. (2022). “Fast and Correct: Context-Aware Code Transformation with Large Language Models”. In: ICSE.
- Beck, Kent (1999). Extreme Programming Explained. Addison-Wesley.
- Blomgren, Michel (2025a). lockd: Distributed Lock Service and Data Platform. [github.com/sa6mwa/lockd](https://github.com/sa6mwa/lockd).
- (2025b). Outcome-Based Agile Framework. [github.com/sa6mwa/obaf](https://github.com/sa6mwa/obaf).
- Brooks, Frederick P (1995). The Mythical Man-Month. Addison-Wesley.
- Burns, Tom and G. M. Stalker (1961). The Management of Innovation. Tavistock Publications.
- Chandler, Alfred D. (1962). Strategy and Structure. MIT Press.
- Chen, Mark et al. (2021). “Evaluating Large Language Models Trained on Code”. In: arXiv preprint arXiv:2107.03374.
- Cockburn, Alistair (2001). Agile Software Development. Addison-Wesley.
- Conway, Melvin E (1968a). “How Do Committees Invent?” In: Datamation 14.4, pp. 28–31.
- Conway, Melvin E. (1968b). “How Do Committees Invent?” In: Datamation 14.4, pp. 28–31.
- Cooke, Roger (1991). Experts in Uncertainty: Opinion and Subjective Probability in Science. Oxford University Press.
- Forsgren, Nicole, Jez Humble, and Gene Kim (2018). Accelerate. IT Revolution Press.
- Galbraith, Jay R. (1973). Designing Complex Organizations. Addison-Wesley.
- Goldratt, Eliyahu M. and Jeff Cox (1984). The Goal: A Process of Ongoing Improvement. North River Press.
- Highsmith, Jim (2001). Agile Software Development Ecosystems. Addison-Wesley.
- Hoek, Andre van der et al. (2016). “Ties that Bind: Coordination in Software Engineering”. In: IEEE Software.
- Humble, Jez and David Farley (2010). Continuous Delivery. Addison-Wesley.
- Humble, Jez, Joanne Molesky, and Barry O'Reilly (2015). Lean Enterprise: How High Performance Organizations Innovate at Scale. O'Reilly Media.
- Hutchins, Edwin (1995). Cognition in the Wild. MIT Press.
- Kerzner, Harold (2017). Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 12th. Wiley.
- Klein, Gary (1998). Sources of Power: How People Make Decisions. MIT Press.
- Knaster, Richard and Dean Leffingwell (2020). SAFe Distilled: Achieving Business Agility with the Scaled Agile Framework. Addison-Wesley Professional.
- Kuusinen, K et al. (2017). “Challenges in Agile Adoption”. In: Proceedings of XP2017.
- Laloux, Frederic (2014). Reinventing Organizations. Nelson Parker.
- Lewallen, David (2020). “Is Agile Dead? A Deep Dive into Agile Failure Modes”. In: Agile Journal.
- Mintzberg, Henry (1979). The Structuring of Organizations. Prentice-Hall.
- Osinga, Frans (2006). Science, Strategy and War: The Strategic Theory of John Boyd. Routledge.
- Qian, Chenxin et al. (2023). “Communicative Agents for Software Engineering”. In: arXiv preprint arXiv:2307.07924.
- Ries, Eric (2011). The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business.
- Robertson, Brian J. (2015). Holacracy: The New Management System for a Rapidly Changing World. Henry Holt and Company.
- Rother, Mike (2010). Toyota Kata. McGraw-Hill.

- Sailer, Kerstin and Ian McCulloh (2020). “Organizing in the Digital Age: Understanding Sociotechnical Interactions”. In: *Journal of Organizational Design*.
- Scaled Agile Inc. (2021). SAFe 5.1 Framework. Tech. rep. Scaled Agile Inc.
- Schwaber, Ken and Jeff Sutherland (2020). The 2020 Scrum Guide. <https://scrumguides.org>.
- Seeber, Isabella et al. (2020). “Machines as Teammates: A Research Agenda on AI in Team Collaboration”. In: *Information & Management*.
- Seeber, Isabella et al. (2020). “Machines as Teammates: A Research Agenda on AI in Team Collaboration”. In: *Information and Management*.
- Senge, Peter M. (1990). *The Fifth Discipline*. Doubleday.
- Shneiderman, Ben (2020). “Human-Centered AI”. In: *Communications of the ACM* 63.10.
- Skelton, Matthew and Manuel Pais (2019). *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. IT Revolution Press.
- Software Productivity Research (2017). “Limitations of Large-Scale Agile Frameworks in Complex Organizations”. In: *SPR Research Note*.
- Trist, Eric (1981). *The Evolution of Socio-Technical Systems*. Occasional Paper.
- Weber, Max (1978). *Economy and Society*. Ed. by Guenther Roth and Claus Wittich. Berkeley: University of California Press.
- Womack, James P. and Daniel T. Jones (1996). *Lean Thinking*. Simon & Schuster.
- Wu, Chunhui et al. (2021). *Management Transformation: Haier’s Rendanheyi Model*. Springer.
- Zhao, W (2023). “A Survey of Multi-Agent Architectures for Autonomous AI Systems”. In: arXiv preprint arXiv:2312.06585.