

TINework

From implicit work to a governed system

Method compendium and critical review

Tinework studies authorized work, turns tacit decisions into candidate rules, and requires accountable policy approval before implementation. The result can be training, configuration, software, bounded AI, or no change.

Executive position

The method is not a new scientific theory. It is an applied synthesis of Jobs to Be Done, contextual inquiry, service and process design, product discovery, improvement science, human-factors research, and AI risk management. Its defensible value is disciplined traceability: observed behavior becomes a candidate rule, an accountable person approves policy, an implemented control remains inspectable, and a measured result can revise the system.

That claim is defensible today. Claims of lower cost, faster delivery, better adoption, or superior outcomes are not yet defensible without repeated engagement data. This paper states both the method and the conditions required to strengthen it.

The sequence

- 01 Define the job
- 02 Specify outcomes
- 03 Collect authorized evidence
- 04 Infer and approve rules
- 05 Implement the smallest sufficient change
- 06 Measure and learn

Audience

Owners and operators whose recurring work is important enough to improve, bounded enough to study, and valuable enough to justify a purpose-built change.

01

The method as currently defined

The method treats the workflow, not the software, as the design object. Each stage must produce an inspectable artifact and a reason to proceed. The table below converts the public method into a decision contract.

Stage	Question	Required output	Pass condition
01 Define the job	Who must accomplish what?	Job statement; executor; circumstance	The statement does not assume a tool or current process.
02 Specify outcomes	What must improve, and for whom?	Measures; baselines; constraints; business value	Each measure has an owner, source, and comparison point.
03 Observe the work	What happens in authorized representative cases?	Actions; context; decisions; exceptions; provenance	Operators confirm the facts and unresolved variation is visible.
04 Infer and approve	What candidate rules explain the decisions?	Candidate rules; corrections; approved policy; authority	Observed behavior, interpretation, and policy remain distinct.
05 Allocate	What is the smallest sufficient intervention?	Skill, policy, workflow, SaaS, software, or AI contracts	Capability, education, risk, authority, and exceptions are explicit.
06 Pilot and learn	Did the change improve the agreed outcome?	Prediction; pilot result; decision record	Evidence supports adjust, adopt, or stop - not automatic scale.

Three operating rules

- A Evidence before design.
Observation and work records constrain the proposed change. A polished map is not evidence that the map is correct.
- B Observed behavior is not policy.
Keep the fact, interpretation, candidate rule, approved policy, and implemented control distinct. A workaround may expose a defect rather than a rule.
- C AI proposes. People authorize.
A consultant or model may assemble evidence and infer a candidate rule. The client retains authority over policy, data, operating roles, workforce effects, pilot exposure, and deployment.

What the synthesis gets right

The value of the method is the connection among disciplines. Jobs to Be Done supplies a solution-independent direction. Observation tests the stated workflow against work as performed. Systems design changes roles and flow before code. Product discovery tests whether a proposed system is valuable, usable, feasible, and viable. Human-factors research warns that automation can leave people with the hardest exceptions. PDSA turns a launch into a prediction and learning cycle. NIST adds continuous AI risk management. [1-9]

- 01

It begins with an outcome, not a feature request.

This keeps a request for an AI agent, dashboard, or integration from becoming the problem definition.
- 02

It distinguishes stated process, observed behavior, and approved policy.

Authorized inquiry and event evidence can expose shadow work, local adaptations, decisions, and exceptions. Operator correction prevents a workaround from becoming a rule by accident. [3,11]
- 03

It makes allocation a design decision.

People, deterministic software, and AI have different failure modes. Explicit allocation is better than treating automation as a default.
- 04

It tests AI at task level.

Generative AI performance varies across tasks. A broad capability claim cannot substitute for evaluation on the actual work. [6,7]
- 05

It separates evidence, inference, policy, and control.

A repeated action or persuasive model output does not authorize policy, adoption, workforce change, or deployment.
- 06

It expects revision.

The design is a candidate until observed results support it. The system remains subject to monitoring, correction, and retirement.

Claim discipline

Defensible now	Not yet established
A coherent, source-grounded method for tracing a workflow change from intent to evidence and decision.	That the method reliably reduces cost, cycle time, error, or implementation risk across clients.
A task-level boundary for deciding whether AI belongs in a workflow.	That Tinework has identified a universal or proprietary formula for allocating work.
A transparent worked example with assumptions and evidence gaps.	A validated client case study or proof of general effectiveness.

03

Red-team review: what is off base

The current method is directionally sound, but several of its clean formulations are stronger than the operating evidence beneath them. These are design defects to correct, not disclaimers to hide.

- 01** The stable job is a hypothesis, not a fact.
 A single job statement can hide competing jobs, several actors, policy obligations, and system effects. Technology can also change what work is possible, so complete solution-independence is unrealistic.
 Stronger rule: Start with a primary job, then map adjacent jobs, affected actors, conflicts, and the boundary conditions under which the statement holds.

- 02** Observation has no sufficiency or participation rule.
 The method does not fully specify case selection, rare exceptions, contradictory evidence, purpose, notice, minimum data, operator correction, or when observation is adequate for a decision.
 Stronger rule: Write an evidence and participation plan first. State the purpose, authorization, cases, records, operators, exceptions, correction path, confidence limits, and missing evidence that can block a conclusion.

- 03** Observed behavior is not policy.
 A repeated action may be sound judgment, a bad workaround, an outdated constraint, or an error. Inferring policy directly from frequency can automate the defect and silence operators.
 Stronger rule: Record the observation, inference, counterexamples, operator correction, accountable approver, approved policy, and implemented control as separate artifacts.

- 04** Desired outcomes can reward the wrong result.
 Speed, effort, and cost are easy to measure but can shift work, degrade quality, increase risk, or consume released capacity without creating value.
 Stronger rule: Pair each outcome with guardrails, a named beneficiary, time to proficiency, human effort, exception and supervision load, an evidence source, a time horizon, and an explicit workforce or capacity plan.

- 05** The allocation model is too coarse.
 Person, deterministic software, or AI is a useful first distinction, not a complete operating model. It omits team coordination, policy, vendor services, user self-service, shared authority, and degrees of automation.
 Stronger rule: Specify a task contract: inputs, outputs, authority, quality threshold, uncertainty, abstention, escalation, recovery, logging, and accountable owner.

03 / CONTINUED

The missing operating disciplines

06 AI-native language creates solution bias.
If AI is part of the identity of the method, teams may feel pressure to find a place for it even when policy, training, deterministic software, or no system is the better intervention.
Stronger rule: Make the method outcome-native. Treat AI as eligible only after the task, evidence, failure cost, and control path are explicit.

07 A pilot can pass while implementation fails.
A bounded test may demonstrate technical performance without adoption, proficiency, integration, fidelity, acceptable cost, or sustained use. Implementation outcomes are distinct from service and business outcomes. [10]
Stronger rule: Measure acceptability, adoption, time to reliable proficiency, appropriateness, feasibility, fidelity, implementation cost, exception load, support demand, reach, and sustainability.

08 Security, privacy, and data readiness enter too late.
Controls appear in the future-state design, but a system can become infeasible before design if identity, permissions, retention, data rights, model exposure, or incident response cannot be satisfied.
Stronger rule: Add an early risk tier and data-rights gate. Carry secure-by-design ownership through development, deployment, maintenance, and retirement. [8,12]

09 The economic claim is a hypothesis.
Custom fit may reduce education and human execution effort. AI may reduce configuration and engineering labor. Neither makes ownership cheap. Support, unique-system training, verification, permissions, vendor drift, and change control can offset the saving.
Stronger rule: Compare no change, policy and training, manual or AI-assisted configuration, SaaS, integration, companion software, replacement, and new software using time to proficiency, human effort, staffing effects, full lifecycle cost, risk, and the value of fit.

10 The method does not yet evaluate itself.
The worked example proves coherence, not effectiveness. There is no evidence yet for repeatability across consultants, agreement among observers, transfer across domains, or outcome improvement across engagements.
Stronger rule: Publish a validation protocol and retain every engagement's baseline, revision history, decision latency, adoption data, operating result, and total cost.

Verdict. Keep the core sequence. Replace its cleanest slogans with explicit contracts. The method becomes stronger when it can say why a job boundary is credible, why the evidence is sufficient, why an allocation is safe, why a pilot is representative, and why an economic claim is earned.

04

Proposed stronger method: version 0.4

The six-stage public method remains a useful summary. The operating method should expand to nine gated stages. Each stage below names the missing requirement and the condition for proceeding.

Stage	New requirement	Gate
00 Qualify	Name the system boundary, sponsor, operators, materiality, risk tier, and why change is needed now.	Do not start when authority, access, or a consequential outcome is absent.
01 Model the jobs	Treat the primary job as a hypothesis. Map adjacent jobs, actors, conflicts, incentives, and external dependencies.	Operators and affected parties agree on the boundary and disagreements are retained.
02 Contract the outcomes	Set outcome, guardrail, adoption, risk, and economic measures with sources, baselines, owners, and time horizons.	No measure is accepted without a collection method and decision use.
03 Plan and collect evidence	Define purpose, notice, authorization, case selection, logs, interviews, observation, exception sampling, minimization, and confidence limits.	Evidence coverage is sufficient for the stated decision, and participants can correct the record.
04 Infer candidate rules	Separate observed fact, interpretation, candidate rule, causal hypothesis, and policy constraint. Test counterexamples with operators.	The candidate explains the evidence better than alternatives and has not been mistaken for policy.
05 Approve policy and alternatives	An accountable owner approves rules, then compares no change, skills, workflow, configuration, extension, replacement, and new software.	The selected intervention wins on outcomes, risk, education, adoption, and total cost.
06 Specify implementation and controls	Link each control to approved policy. State desired state, authority, tests, failure threshold, monitoring, recovery, rollback, and owner.	A person can review the change, detect failure, recover safely, and stop the system.
07 Pilot adoption and performance	Pre-register prediction, exposure, stop rules, participant set, learning period, adoption measures, and decision criteria.	The result is compared with baseline and guardrails before wider use.
08 Operate, review, and retire	Assign service ownership, security, monitoring, change control, incident response, cost review, and retirement criteria.	The system has a funded owner and a dated review or retirement decision.

The change is structural. It moves participation, inference, policy authority, education, workforce effects, security, economics, and operation into the method. It makes no change legitimate and custom work the last sufficient layer.

05

The six contracts every engagement should leave behind

These contracts turn the method from persuasive analysis into auditable operating practice.

- | | |
|----|--|
| 01 | Boundary contract.
Primary and adjacent jobs; actors; sponsor; operators; included systems; excluded systems; constraints; materiality; decision authority; risk tier. |
| 02 | Evidence contract.
Purpose; authorization; case selection; observations; event records; interviews; exceptions; provenance; minimization; correction rights; contradictions; coverage; confidence; unresolved gaps. |
| 03 | Outcome and economics contract.
User and business outcomes; guardrails; baseline; time to proficiency; human effort; task coverage; exceptions; supervision; staffing effects; delivery and lifecycle cost; capacity destination. |
| 04 | Policy and control contract.
Observed behavior; interpretation; candidate rule; counterexamples; approved policy; accountable owner; implemented control; authority; audit event; recovery. |
| 05 | Pilot and adoption contract.
Prediction; comparison; exposure; participants; duration; stop rules; adverse events; adoption; proficiency; support; exception and supervision load; workforce effects; authorized next step. |
| 06 | Operation and retirement contract.
Service owner; support level; monitoring; incident response; model or dependency changes; cost review; outcome review; rollback; portability; retention; retirement trigger. |

Non-negotiable gates

- G1. No policy from observed behavior until operators can correct the interpretation and an accountable owner approves the rule.
- G2. No automation until the current work, its exceptions, and the function of each proposed removal are understood.
- G3. No configuration write until the current and desired states, approval, test evidence, owner, and rollback path are recorded.
- G4. No AI until the exact task, evaluation set, failure cost, human authority, and recovery path are specified.
- G5. No pilot until the prediction, baseline, guardrails, exposure, and stop rules are recorded.
- G6. No wider use until performance, adoption, security, and operating ownership pass their separate gates.
- G7. No savings claim until full lifecycle cost and observed business value are measured over a stated period.
- G8. No workforce claim until task coverage, exceptions, supervision, role effects, and the destination of released capacity are explicit.

06

How to position and prove the method

Do not market Tinework as a proprietary methodology that guarantees transformation. Market it as evidence-bound execution for one consequential workflow: observe authorized work, expose the decisions, infer candidate rules, approve policy, implement the smallest sufficient change, and measure what happens next.

The commercial promise

A client should leave with a verified current-state model, explicit outcomes and guardrails, candidate rules, approved policy, compared alternatives, and a task-level future-state design. The implementation may be training, configuration, software, bounded AI, or no change. Every implemented control needs tests, an owner, and a recovery path.

The validation agenda

- | | |
|----|---|
| 01 | Current-state reliability.
Measure time to an operator-confirmed map, number of material corrections after confirmation, and agreement among operators and observers. |
| 02 | Decision quality.
Track unresolved questions at each gate, alternatives considered, reasons for the selected design, and reversals caused by missing evidence. |
| 03 | Pilot validity.
Compare predicted and observed outcomes, guardrail breaches, exception rates, adoption, proficiency, and the decision made from the result. |
| 04 | Sustained operation.
Measure use, fidelity, outcome performance, incidents, support demand, model or dependency drift, and retirement or revision decisions after launch. |
| 05 | Economics.
Report time to proficiency, human effort per case, task coverage, exceptions, supervision, staffing and hiring effects, partner fees, delivery, support, maintenance, switching cost, realized benefit, and where released capacity went. |

Claim ladder

Now: describe the method, its source lineages, its artifacts, and its limits. After repeated engagements: publish descriptive benchmarks and failure patterns. After comparable baselines and follow-up: make bounded outcome and economic claims for the contexts actually observed.

The strongest version of Tinework is not the one that claims certainty. It is the one that makes uncertainty, authority, evidence, and revision operational.

ADDENDUM

What an SMB buyer can reasonably reject

An objection is evidence about exposure. The right response is not always a rebuttal. Tinework should answer a misunderstanding, redesign an avoidable risk, or respect a no that the method cannot safely overcome.

Buyer question	Method requirement
Why should this workflow change now?	A qualification gate must establish materiality, urgency, authority, operator access, and a bounded decision.
Why not fix it internally, configure what we own, buy software, or do nothing?	The method must compare policy, training, internal work, manual or AI-assisted configuration, vendor software, integration, custom work, and no-build before delivery is authorized.
Are you watching employees or preparing to replace them?	Observation must have a stated purpose, authorization, minimum data, operator correction rights, and separate policy authority. Workforce change needs explicit task, exception, supervision, and impact evidence.
What cash, attention, and operating risk are we accepting?	Diagnostic, pilot, and production commitments need separate prices, exposure caps, client-owned outputs, stop rights, and full lifecycle costs.
Why should we trust the provider as well as the design?	Legal identity, insurance, security practices, data terms, support capacity, conflicts, domain authority, and evidence of prior results must be inspectable.
What happens if the system fails, a dependency changes, or Tinework is unavailable?	Client-controlled repositories and credentials, monitoring, rollback, backup maintenance, portable data, transition rights, and retirement criteria must exist before launch.

The current provider gap

The method is more developed than Tinework's provider assurance. Before broad marketing, the practice needs verified client evidence, a continuity arrangement beyond one person, a vendor assurance pack, clear legal and intellectual-property terms, defined support capacity, and independent controls on the incentive to recommend a build.

The no-sale rule

Do not proceed when the outcome is immaterial, authority or operator access is absent, evidence cannot be handled lawfully, failure cannot be bounded, no operating owner exists, a safer alternative wins, or Tinework cannot satisfy a material provider requirement.

A separate objection compendium expands these questions into thirty-four customer objections and twelve required process changes.

SOURCES

Selected references

The method is a Tinework synthesis. Inclusion of a source means that it shaped a specific part of the method; it does not imply endorsement of the synthesis by the author or institution.

1. Ulwick, A. W. Jobs to Be Done: Theory to Practice. Strategyn, 2016. <https://strategyn.com/jobs-to-be-done/>
2. Bettencourt, L. A., and Ulwick, A. W. 'The Customer-Centered Innovation Map.' Harvard Business Review, 2008. <https://hbr.org/2008/05/the-customer-centered-innovation-map>
3. Beyer, H., and Holtzblatt, K. Contextual Design. Morgan Kaufmann, 1997. <https://shop.elsevier.com/books/contextual-design/holtzblatt/978-0-08-050304-2>
4. Shostack, G. L. 'Designing Services That Deliver.' Harvard Business Review, 1984. <https://store.hbr.org/product/designing-services-that-deliver/84115>
5. Hammer, M. 'Reengineering Work: Don't Automate, Obliterate.' Harvard Business Review, 1990. <https://hbr.org/1990/07/reengineering-work-dont-automate-obliterate>
6. Bainbridge, L. 'Ironies of Automation.' Automatica 19(6), 1983. [https://doi.org/10.1016/0005-1098\(83\)90046-8](https://doi.org/10.1016/0005-1098(83)90046-8)
7. Dell'Acqua, F., et al. 'Navigating the Jagged Technological Frontier.' Harvard Business School Working Paper 24-013, 2023. <https://www.hbs.edu/ris/download.aspx?name=24-013.pdf>
8. Tabassi, E. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1, 2023. <https://doi.org/10.6028/NIST.AI.100-1>
9. W. Edwards Deming Institute. 'PDSA Cycle.' <https://deming.org/explore/pdsa/>
10. Proctor, E., et al. 'Outcomes for Implementation Research: Conceptual Distinctions, Measurement Challenges, and Research Agenda.' Administration and Policy in Mental Health 38, 2011. <https://doi.org/10.1007/s10488-010-0319-7>
11. IEEE Task Force on Process Mining. Process Mining Manifesto. 2012. <https://www.tf-pm.org/resources/manifesto>
12. CISA, NSA, FBI, and international partners. Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Security-by-Design and -Default. 2023. <https://www.cisa.gov/resources-tools/resources/secure-by-design>
13. Cagan, M. 'The Four Big Risks.' Silicon Valley Product Group, 2017. <https://www.svpg.com/four-big-risks/>
14. Object Management Group. Business Process Model and Notation, Version 2.0.2. <https://www.omg.org/spec/BPMN/2.0/>

Document status

Design candidate. This compendium defines the current public method, challenges its unsupported assumptions, and proposes a stronger operating version. It is not a certification, standard, or claim of validated client outcomes.

tinework.com | rcerf@lecerf.ai