A Tiered Maturity Model for Cost-Benefit Analysis of Agentic versus Traditional Software Delivery

Brooks Johnson
AI Strategy Enterprise Consulting

Abstract
The arrival of agentic coding, in which a large language model orchestrates code authoring, test execution, and iterative refinement under human supervision, has created a measurement problem for organizations that must justify, price, and govern software work. Traditional cost estimation methods were calibrated for human teams operating under iterative frameworks such as Scrum, and they do not naturally accommodate a delivery mode in which most authoring labor is performed by an agent and most human labor shifts toward specification, review, and acceptance. This paper proposes a three-tier maturity model for performing cost-benefit analysis (CBA) of agentic versus traditional software delivery. Tier 1, the retrospective tier, evaluates completed agentic work against a counterfactual Scrum baseline using fully loaded role costs. Tier 2, the project-inception tier, applies the same evaluation logic at the start of a project whose scope is well defined, drawing on reference-class data from Tier 1 outcomes. Tier 3, the consultative engagement tier, extends the model to early-stage opportunities in which scope itself carries uncertainty, expressing both pathways as distributions rather than point estimates. We argue that each tier delivers usable value at its own level of organizational maturity, that the tiers dovetail across the project lifecycle, and that agentic sizing, properly disciplined, can serve as a unifying instrument for sales, pricing, and execution governance.

Keywords: agentic coding, cost-benefit analysis, software cost estimation, reference class forecasting, agile delivery, capability maturity, sales engineering, professional services

1. Introduction

For roughly two decades, software cost estimation in the enterprise has been an exercise in calibrating human throughput. Function points, story points, COCOMO II, and various proprietary parametric methods all share a common premise: the unit of work is mediated by a person, and the principal cost lever is the size and skill mix of the team performing it (Boehm et al., 2000; Cohn, 2006). Agile delivery refined this premise rather than displacing it. Although Scrum and its relatives reorganized planning around small iterations and emergent design, they still measured work in person-time, and they still carried meaningful overhead in the form of product management, scrum mastery, technical leadership, and other coordination roles.

Agentic coding disrupts this premise. In an agentic delivery model, a language model with tool access, often supervised by a single engineer, performs a substantial share of authoring, refactoring, and verification work that previously required multiple humans. Tools such as Claude Code, Cursor, and GitHub Copilot occupy different points on the spectrum from in-IDE assistance to autonomous task execution, but they share the property that the marginal unit of labor is no longer well captured by a person-hour. The cost of work becomes a composite of token consumption, agent runtime, and human supervisory effort, while the schedule becomes a function of how rapidly the supervising human can specify and adjudicate output.

This shift is not a marginal adjustment to estimation practice. It changes what is being counted. An organization that wants to make defensible decisions about when to use agentic methods, how to price agentic work for clients, and how to set internal budgets for agentic initiatives needs a

measurement framework that is at least as rigorous as the one it inherited from agile planning, while accommodating the new cost structure.

The contribution of this paper is a three-tier maturity model for cost-benefit analysis of agentic versus traditional software delivery. Each tier is associated with a stage of project knowledge: completed work, well-scoped work, and consultatively scoped work. Each tier offers a defensible analytical approach calibrated to the information available at that stage. We describe the methodology, the data inputs, an illustrative formula, and a worked example for each tier. We then show how the tiers dovetail across the project lifecycle so that estimates produced at the consultative stage can be refined as scope clarifies and validated retrospectively as work completes. The implication is that a single estimation discipline can serve sales, sizing, and execution governance, which is rarely true of the methods it would replace.

The remainder of the paper is organized as follows. Section 2 reviews related work in software cost estimation and the early literature on agentic delivery. Section 3 introduces the maturity model. Sections 4, 5, and 6 develop Tiers 1, 2, and 3 respectively. Section 7 discusses how the tiers dovetail across the lifecycle. Section 8 examines the value and tradeoffs of the framework. Section 9 concludes.

2. Background and Related Work

Parametric software cost estimation has a long lineage. COCOMO and its successor COCOMO II calibrated effort to the size of the system, the experience of the team, and a set of environmental modifiers (Boehm et al., 2000). Function point analysis, codified through IFPUG and related bodies, sized work by counting externally observable system behaviors (Albrecht, 1979; Garmus and Herron, 2001). These methods retain explanatory power for large systems with stable requirements but tend to require investments in calibration and historical data that many organizations cannot sustain.

Agile estimation displaced parametric methods in many organizations not because it was more accurate, but because it produced usable forecasts faster. Story points and velocity track relative complexity per sprint, and reference class forecasting techniques (Flyvbjerg, 2006) can be applied to convert that history into schedule and cost projections. The principal cost driver in an agile shop is team composition, and a fully loaded Scrum team typically includes engineering staff, a product owner, a scrum master, quality assurance, and a proportionate share of architecture, design, and management overhead. Industry surveys consistently place these support roles between thirty and fifty percent of total team cost (Standish Group, 2020; Project Management Institute, 2021).

Agentic coding is too new to have generated a mature estimation literature. Early studies have focused on productivity effects of in-IDE assistants. Peng et al. (2023) reported substantial speed gains for routine tasks using Copilot. Subsequent industry reports have explored autonomous coding agents, with results that range from modest acceleration to multiple-fold compression of cycle time depending on task type and supervision practice (GitHub, 2023; McKinsey, 2023). What is missing from this literature, and what motivates the present paper, is a disciplined framework for comparing agentic delivery to its traditional counterpart on a project-by-project basis with attention to the full cost stack, not merely the engineer-hour.

3. A Tiered Maturity Model

We propose that organizations adopting agentic coding progress through three tiers of cost-benefit analytical maturity. The tiers are distinguished by the state of knowledge at the time the analysis is performed and by the techniques required to handle the uncertainty at that state. Figure 1 illustrates the model.

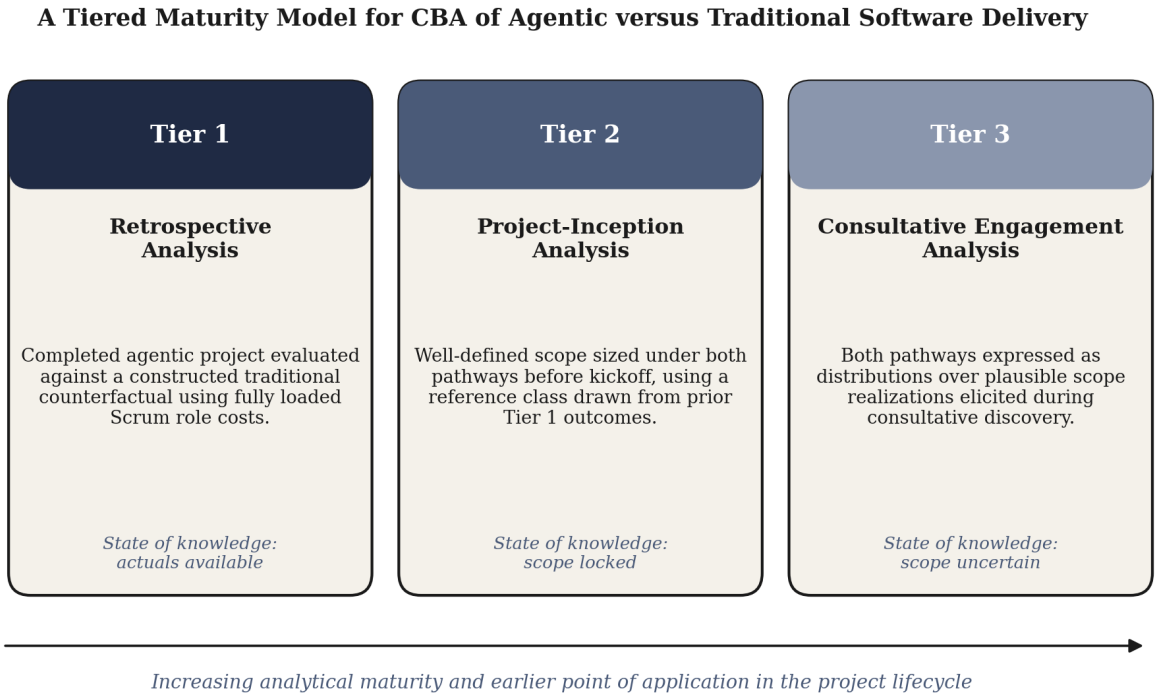

Figure 1. A three-tier maturity model for cost-benefit analysis of agentic versus traditional software delivery. Tier 1 evaluates completed work; Tier 2 evaluates well-scoped work prior to kickoff; Tier 3 evaluates consultatively scoped opportunities with explicit treatment of scope uncertainty.

Tier 1, the retrospective tier, is the entry point. After an agentic project completes, its actual costs are tallied and compared to a counterfactual traditional baseline constructed from typical Scrum role compositions and prevailing fully loaded rates. The output is a delta, expressed in cost and schedule, that captures the realized benefit of the agentic approach for that specific engagement.

Tier 2, the project-inception tier, applies the same comparative logic at the start of a project whose scope is well defined and ready for execution. The traditional baseline is constructed from the team that would have been mobilized; the agentic estimate is constructed from reference-class data accumulated through Tier 1 history. The output is a point estimate accompanied by uncertainty bounds.

Tier 3, the consultative engagement tier, extends the analysis to opportunities in which the scope itself has not yet been fully defined. In a consultative engagement, requirements are elaborated through structured discovery, and meaningful uncertainty remains about the size and shape of the

work. At this tier, both pathways are expressed as distributions over plausible scope realizations, and the cost-benefit comparison becomes a comparison of distributions rather than scalars.

Each tier produces decision-useful output at its level of certainty. An organization need not reach Tier 3 to benefit from the framework. However, organizations that progress through all three tiers gain the ability to dovetail estimates across the project lifecycle, refining initial consultative analyses as scope clarifies and validating final delivery against the original business case.

4. Tier 1: Retrospective Cost-Benefit Analysis

4.1 Definition and Purpose

Tier 1 analysis is performed after an agentic project has completed. Its purpose is to answer a single question with the highest possible fidelity: given what we now know about what this project actually cost and produced under agentic delivery, what would it have cost and how long would it have taken under a traditional agile model? The output supports portfolio decisions, retrospective communications to executive stakeholders, and the construction of a reference class for later Tier 2 and Tier 3 analyses.

4.2 Data Inputs

Tier 1 analysis requires complete actuals for the agentic engagement and a defensible traditional counterfactual. The agentic actuals comprise four categories. The first is direct platform cost, principally token consumption and any per-task fees levied by the agentic platform. The second is compute infrastructure cost, including any sandboxed environments or build runners that supported agent execution. The third is human supervisory cost, which includes the time spent by engineers specifying tasks, reviewing output, performing acceptance, and intervening when the agent stalls. The fourth is overhead allocation, including the share of platform engineering, security review, and management attention that the engagement consumed.

The traditional counterfactual requires constructing a plausible Scrum team for the same scope. This means specifying team size and role composition, sprint length, expected velocity, and fully loaded labor rates including benefits, facilities, and a defensible share of organizational overhead. Where reasonable, the counterfactual should be calibrated against an analogous project that the organization actually delivered in the traditional mode.

4.3 Illustrative Formula

Let $\boldsymbol{C}_{\mathrm{A}}$ denote the total cost of the agentic engagement and $\boldsymbol{C}_{\mathrm{T}}$ the total cost of the traditional counterfactual. Each is decomposed as follows:

$$C_A = C_{tokens} + C_{compute} + (h_{sup} \cdot r_{sup}) + O_A$$

$$C_T = \sum_{i \in R} (h_i \cdot r_i) + O_T$$

In the agentic expression, h_{sup} is supervisory hours and r_{sup} is the loaded rate of supervising engineers; O_A is the allocated overhead for the agentic engagement. In the traditional expression, R is the set of roles on the counterfactual Scrum team, h_i and r_i are hours and loaded rate for role i, and O_t is the allocated overhead. The retrospective delta is given by:

$$\Delta = C_T - C_A$$

The schedule delta is computed analogously by comparing realized agentic duration to the counterfactual project duration as a function of team velocity and scope.

4.4 Worked Example

Consider a completed agentic engagement that delivered a customer-facing internal data ingestion service over a three-week elapsed period. Table 1 summarizes the actuals and the counterfactual.

Table 1. Tier 1 worked example: agentic actuals and traditional counterfactual.

Cost component	Agentic actuals	Traditional counterfactual
Platform tokens and compute	$4,800	
Supervisory engineering (90 h at $185 loaded)	$16,650	
Allocated overhead	$5,500	
Engineering staff (2.5 FTE x 9 wk)		$155,000
Product owner (0.4 FTE x 9 wk)		$24,300
Scrum master (0.3 FTE x 9 wk)		$17,800
Quality assurance (0.6 FTE x 9 wk)		$34,000
Architecture and design (0.2 FTE x 9 wk)		$14,400
Management overhead allocation		$14,450
Subtotal	**$26,950**	**$259,950**
Realized cost delta		**$233,000**
Schedule	3 weeks elapsed	9 weeks expected

The retrospective delta in this example is approximately $233,000, with a schedule compression of roughly seven weeks. The Tier 1 finding is concrete and defensible because both sides are anchored in observed or directly inferable data.

4.5 Limitations

Tier 1 has three principal limitations. First, the traditional counterfactual is, by definition, a hypothetical, and reasonable practitioners can construct different team compositions for the same scope. Sensitivity analysis on the team composition is therefore essential. Second, the agentic actuals can underrepresent hidden costs such as residual technical debt or test gaps that surface only after deployment; these costs should be amortized into the analysis where possible. Third, Tier 1 produces a finding about one engagement and not a generalization. Aggregating across many Tier 1 findings, with attention to similarity in scope and complexity, is what produces the reference class that enables Tier 2.

5. Tier 2: Project-Inception Cost-Benefit Analysis

5.1 Definition and Purpose

Tier 2 analysis is performed at the start of a project whose scope is well defined and ready for execution. The defining condition is that the work has been decomposed into a set of artifacts, behaviors, or epics whose sizing is no longer fundamentally uncertain. The purpose of the analysis is to inform the choice of delivery model and to set a defensible budget and schedule for whichever pathway is selected.

5.2 Methodology

Tier 2 requires two estimates of the same well-scoped work. The traditional estimate is constructed using whatever method the organization already trusts for agile projects, typically a combination of relative sizing, reference velocity, and a team composition consistent with the organization's standard practice. The agentic estimate is constructed from reference-class data accumulated through Tier 1 analyses of analogous prior work. The reference class is defined by the scope characteristics of the new project: domain, complexity, integration surface, regulatory burden, and so on. Where reference data is thin, the practitioner can decompose the new scope into components for which reference data does exist.

The output of Tier 2 is a point estimate for each pathway accompanied by uncertainty bounds derived from the variance observed in the reference class. The recommended practice is to report the comparison as a probability statement: for example, under reference-class assumptions, the agentic pathway is expected to cost between sixty and seventy-five percent of the traditional pathway with eighty percent confidence.

5.3 Illustrative Formula

Using reference-class forecasting, the estimated costs under each pathway for a well-defined scope are:

$$\hat{C}_A(S) = \sum_{j \in J} n_j(S) \cdot \bar{c}_j^A$$

$$\hat{C}_T(S) = \sum_{j \in J} n_j(S) \cdot \bar{c}_j^T$$

where J is a set of scope component types (for example, CRUD endpoints, integration adapters, data pipelines, UI screens), $n_j(S)$ is the count of components of type j in scope S, and the bar-c terms are the mean realized costs per component type observed in the agentic and traditional reference classes respectively. Uncertainty bounds are derived from the within-class standard deviation.

5.4 Worked Example

A logistics firm proposes to build an order-routing dashboard with seven CRUD endpoints, two integration adapters to existing systems, three data pipelines, and four UI screens. The reference class, drawn from twelve completed analogous engagements (eight agentic, four traditional), produces per-component cost means and standard deviations. Table 2 shows the resulting Tier 2 estimate.

Table 2. Tier 2 worked example: reference-class projection for a well-scoped dashboard.

Component type	Count	Agentic per unit	Traditional per unit	Agentic subtotal	Traditional subtotal
CRUD endpoint	7	$4,200	$14,800	$29,400	$103,600
Integration adapter	2	$18,500	$44,000	$37,000	$88,000
Data pipeline	3	$22,000	$58,000	$66,000	$174,000
UI screen	4	$13,400	$33,200	$53,600	$132,800
Total point estimate				**$186,000**	**$498,400**
80% confidence interval				$158K - $221K	$441K - $562K

The Tier 2 analysis projects the agentic pathway at approximately $186,000 against a traditional pathway of approximately $498,000. Because the eighty percent confidence intervals do not overlap, the analysis supports a confident recommendation in favor of the agentic pathway, with the explicit caveat that the scope is in fact representative of the reference class.

5.5 Limitations

Tier 2 inherits the quality of its reference class. An organization with only one or two Tier 1 outcomes cannot perform credible Tier 2 analyses, although it can perform pilot Tier 2 analyses that explicitly flag their reference-class thinness. The most common error in Tier 2 is reference-class mismatch, in which a new project is forced into a reference class whose characteristics it does not actually share. Practitioners should be explicit about the basis for inclusion and should hold out the analysis when the new project sits outside the supported envelope.

6. Tier 3: Consultative Engagement Cost-Benefit Analysis

6.1 Definition and Purpose

Tier 3 analysis is performed during a consultative engagement, before scope has been fully elaborated. The defining condition is that the project carries dimensions of genuine uncertainty: the customer may not yet know what they want, the integration surface may not be mapped, the regulatory posture may be unresolved, or the business outcome may admit multiple technical

realizations. Tier 3 is the analytical setting for sales engineering, advisory work, and any engagement in which the cost-benefit conversation must precede the scope-fixing conversation.

6.2 Methodology

Tier 3 expresses both pathways as distributions over plausible scope realizations rather than as point estimates. The practitioner identifies the principal dimensions of scope uncertainty (for example, the number of integrations, the depth of compliance evidence required, the breadth of supported user personas), assigns plausible ranges to each, and propagates the resulting scope distribution through the Tier 2 reference-class machinery for each pathway. The output is a pair of distributions, one for the agentic pathway and one for the traditional pathway, supporting a probabilistic comparison.

The recommended technique is straightforward Monte Carlo sampling. The practitioner draws a scope realization, evaluates the Tier 2 expression for both pathways, and repeats for several thousand iterations. The distributions can be summarized by their means, medians, and key percentiles, and the comparison can be reported as the probability that the agentic pathway costs less than a specified fraction of the traditional pathway, along with the expected savings and the value at risk if the scope realizes at the upper bound.

6.3 Illustrative Formula

Let the script S denote the distribution over plausible scope realizations elicited during consultation. For each draw from that distribution, evaluate the Tier 2 expressions for both pathways and accumulate the resulting cost distributions:

$$F_A(c) = \Pr\left(\hat{C}_A(S) \le c \mid S \sim \mathcal{S}\right)$$

$$F_T(c) = \Pr\left(\hat{C}_T(S) \le c \mid S \sim \mathcal{S}\right)$$

The decision-useful summary statistics include the expected delta and the probability of savings:

$$\mathbb{E}[\hat{C}_T(S) - \hat{C}_A(S)] \,, \quad \Pr(\hat{C}_A(S) < \hat{C}_T(S))$$

6.4 Worked Example

A mid-market manufacturer engages the consulting practice to scope an analytics platform for its production lines. The scope dimensions elicited during consultation include the number of production lines (between four and twelve), the depth of historical data integration (a binary dimension corresponding to whether the customer can deliver clean historical extracts), the number of dashboard personas (between two and five), and the compliance posture (whether

export-control evidence is required). Monte Carlo sampling over the elicited ranges, parameterized by the reference class, produces the distribution summarized in Table 3.

Table 3. Tier 3 worked example: Monte Carlo distribution summary across elicited scope.

Statistic	Agentic pathway	Traditional pathway
Expected cost (mean)	$510,000	$1,380,000
10th percentile	$310,000	$880,000
50th percentile (median)	$495,000	$1,340,000
90th percentile (value at risk)	$720,000	$1,960,000
Probability agentic costs less than half of traditional	**78%**	
Expected savings (traditional minus agentic)	**$870,000**	

The Tier 3 analysis indicates that the agentic pathway is expected to cost between $310,000 and $720,000 across the elicited scope distribution, against a traditional pathway expected to cost between $880,000 and $1,960,000. The probability that the agentic pathway will cost less than half of the traditional pathway is approximately seventy-eight percent. The value at risk for the agentic pathway at the ninetieth percentile is approximately $640,000, which serves as a defensible not-to-exceed for an initial commercial proposal.

6.5 Limitations

Tier 3 is the most demanding tier and the most exposed to model risk. The elicited scope distribution is itself an artifact of the consultative process, and its quality reflects the discipline of the engagement team. Practitioners should treat the elicited distribution as a hypothesis, document the assumptions explicitly, and revisit them as the engagement progresses. The Monte Carlo machinery does not produce certainty; it produces a structured representation of the uncertainty that already existed.

7. Dovetailing the Tiers Across the Project Lifecycle

The three tiers are not alternatives. They are stages of a single estimation discipline that follows a project from first contact to retrospective review. Figure 2 illustrates the dovetailing.

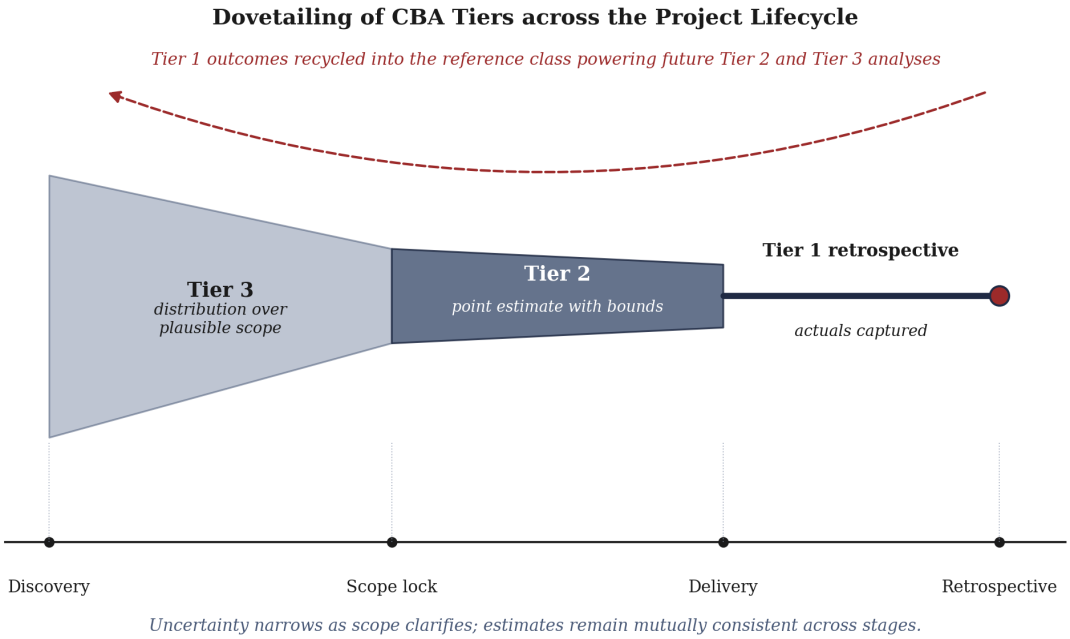

Figure 2. Dovetailing of the three CBA tiers across the project lifecycle. A consultative Tier 3 distribution narrows into a Tier 2 point estimate with bounds as scope clarifies, which is then validated by a Tier 1 retrospective. Tier 1 outcomes are recycled into the reference class that powers future Tier 2 and Tier 3 analyses.

The dovetail operates in two directions. Moving forward through the lifecycle, the Tier 3 distribution narrows as scope clarifies during discovery and elaboration. By the time the project is ready for execution, the practitioner has a Tier 2 estimate; by the time the project completes, the practitioner has a Tier 1 retrospective. Moving backward, Tier 1 outcomes are aggregated into the reference class that calibrates future Tier 2 and Tier 3 analyses, which closes the loop. An organization that maintains this discipline accumulates a compounding asset: each completed project sharpens the estimation accuracy for every future project that resembles it.

The dovetail also supports active cost control during execution. If an engagement opened with a Tier 3 distribution centered at, for example, $500,000, and Tier 2 refinement at scope-lock produced a point estimate of $420,000, the project can be governed against the Tier 2 estimate while retaining the Tier 3 envelope as a contractual ceiling. Variance against the Tier 2 estimate becomes an early warning indicator that triggers managerial review, while breaches of the Tier 3 envelope trigger commercial renegotiation. The same machinery thus serves sales, planning, and governance with internally consistent numbers.

8. Discussion: Value and Tradeoffs

8.1 Value Across the Commercial Lifecycle

The principal claim advanced in this paper is that agentic sizing, properly disciplined through the tiered framework, is a unifying instrument across sales, sizing, and execution. The traditional alternative is a patchwork. Sales teams produce rough-order-of-magnitude estimates from intuition

or analog deals. Delivery teams reestimate at kickoff using a different method, often disconnected from the sales artifact. Finance reports on actuals using a third basis. The disconnect produces commercial risk, internal friction, and limited organizational learning.

A tiered CBA discipline supplies a single estimation backbone. Sales engineers use Tier 3 to produce probabilistic commercial proposals. Delivery managers use Tier 2 to plan and budget. Finance and engineering leadership use Tier 1 to evaluate outcomes and recycle learnings. Because each tier is mathematically connected to the next, variance between sales and delivery becomes diagnostic rather than political. If a Tier 2 estimate diverges materially from the Tier 3 distribution at scope-lock, the cause is either a substantive scope movement (commercially actionable) or a calibration error (operationally actionable). Either way, the conversation is grounded.

8.2 Tradeoffs and Failure Modes

The framework is not free. Three categories of cost should be anticipated. The first is the analytical overhead of maintaining the reference class. An organization with a thin reference class will produce wide uncertainty bounds and may find Tier 2 and Tier 3 results less actionable than expected. Investment in disciplined Tier 1 capture is a precondition for downstream value.

The second category is the risk of reference-class drift. Agentic tooling is improving rapidly, and per-component costs observed in 2025 may not predict per-component costs in 2027. Practitioners should weight recent observations more heavily and should periodically audit the reference class for staleness. A useful practice is to recalibrate every two quarters or after any platform change that materially affects unit economics.

The third category is organizational. The tiered framework asks sales, delivery, and finance to use a shared analytical artifact. Organizations whose internal incentives reward disconnection between these functions will resist the framework, and the resistance will be more difficult to address than the technical work. Adoption typically proceeds best when sponsored at the executive level and piloted on a small portfolio of representative projects before scaling.

8.3 Boundaries of the Framework

The framework is silent on questions of quality, risk, and strategic fit. It compares cost and schedule between two delivery pathways. It does not, in its current form, value the option to learn faster, the strategic implications of in-house agentic capability, or the residual risk associated with agent-mediated authoring under regulatory scrutiny. These considerations should accompany any tiered CBA output as qualitative supplements; they should not be silently absorbed into the cost figures because doing so erodes the diagnostic value of the comparison.

9. Conclusion

The estimation problem that agentic coding poses is not solved by adapting existing parametric methods. It is solved by recognizing that the question of whether to use agentic delivery, what to charge for it, and how to govern it requires a single estimation discipline that operates at three distinct levels of project knowledge. Tier 1 evaluates completed work against a traditional counterfactual. Tier 2 evaluates well-scoped work prior to kickoff using a reference class derived from Tier 1. Tier 3 evaluates consultatively scoped opportunities using a distribution over plausible scope realizations evaluated through the Tier 2 machinery. The tiers are mutually reinforcing, and an organization that adopts all three accumulates an estimation asset that compounds over time.

The broader implication is that agentic sizing, treated rigorously, becomes a strategic instrument rather than a technical artifact. It allows the sales conversation, the delivery plan, and the executive review to share a common analytical foundation. Where traditional estimation methods reproduced the fragmentation of the organization, the tiered framework offered here proposes a small but consequential alternative: that the same numbers, refined as the project progresses, can serve every stakeholder who needs to act on them.

References

Albrecht, A. J. (1979). Measuring application development productivity. Proceedings of the IBM Applications Development Symposium, 83-92.

Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., Reifer, D. J., and Steece, B. (2000). Software Cost Estimation with COCOMO II. Prentice Hall.

Cohn, M. (2006). Agile Estimating and Planning. Prentice Hall.

Flyvbjerg, B. (2006). From Nobel prize to project management: Getting risks right. Project Management Journal, 37(3), 5-15.

Garmus, D. and Herron, D. (2001). Function Point Analysis: Measurement Practices for Successful Software Projects. Addison-Wesley.

GitHub (2023). Research: Quantifying GitHub Copilot's impact on developer productivity. GitHub Engineering Reports.

McKinsey and Company (2023). Unleashing developer productivity with generative AI. McKinsey Digital.

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv preprint arXiv:2302.06590.

Project Management Institute (2021). Pulse of the Profession 2021: Beyond Agility. PMI Research.

Standish Group (2020). CHAOS Report 2020: Beyond Infinity. The Standish Group International.