PART I · THE ARGUMENT

The Flow

OPENING CASE

The budget that ran out in April

DATED: DECEMBER 2025 TO MAY 2026. FIGURES AND DATES WERE CHECKED AGAINST THE CITED REPORTING. THE PRIMARY ACCOUNT BEHIND THAT REPORTING REMAINS PAYWALLED. SEE THE SOURCE REGISTER.

By April 2026, just four months into the year, Uber had already exhausted its annual AI budget.[1] Nothing had broken. No contract had been renegotiated, and no vendor had raised its prices. Engineers were simply doing the work the company had asked them to do by using the tool it had given them.

The tool was Claude Code, an AI assistant that writes and edits software. Uber had begun rolling it out to roughly 5,000 engineers in December 2025.[2] Adoption was rapid. By February, 32 percent of the engineering organization was using it. By March, that share had reached 84 percent.[3]

By adoption measures, the rollout looked like a success. Engineers adopted the tool quickly and used it heavily. Macdonald said that a quarter of the company's code commits in the previous quarter had come through Claude Code.[4] But those measures establish adoption, not value. That distinction is central to this chapter.

---

1. Janakiram MSV, "Uber burns its 2026 AI budget in four months on Claude Code," *Forbes*, May 17, 2026, accessed August 30, 2026. Reported timing of the budget exhaustion.
2. Janakiram MSV, "Uber burns its 2026 AI budget in four months on Claude Code," *Forbes*, May 17, 2026, accessed August 30, 2026. Engineer count and rollout timing.
3. Janakiram MSV, "Uber burns its 2026 AI budget in four months on Claude Code," *Forbes*, May 17, 2026, accessed August 30, 2026. Reported adoption in February and March.
4. Andrew Macdonald, "Uber's swerve on gas prices, hotels, a driverless future," *Rapid Response, Masters of Scale*, May 2026. Share of code commits, in the executive's own words.

Public reporting does not explain how Uber calculated its AI budget. The outcome, however, is consistent with the way companies have long budgeted for software: multiply the annual price per user by the expected number of users and treat that total as the year's cost. For conventional software, that method generally works.

This method endured because the assumptions behind a seat-based forecast can be checked before the year begins. The annual price is known, and the HR system already holds the headcount. Multiply one by the other, and finance has a number it can explain and defend in a planning meeting.

The cost per engineer, however, was not fixed. Reported monthly spending ran from $150 to $250 for the average engineer and from $500 to $2,000 for the heaviest users.[5] The price had not changed. Uber was paying for consumption, so wider adoption and heavier use drove spending higher.

What matters is less the size of the overrun than how it happened. Uber did not simply miss its budget and discover the variance at year-end. Spending had been outrunning the plan for months, yet no one appears to have noticed until the annual budget was gone. A forecast that is never tested against actual spending is not a forecast. It is a hope with a number attached.

A useful February review would have placed actual consumption and the pace of adoption beside the annual plan. Someone responsible for the budget would then have needed to compare them. Public reporting does not reveal whether Uber had such a view or what its finance team could see. It establishes only the result: nothing changed the course of spending before the money was gone.

---

5. Janakiram MSV, "Uber burns its 2026 AI budget in four months on Claude Code," *Forbes*, May 17, 2026, accessed August 30, 2026. Reported per-engineer monthly cost figures, verified against the article at Stage 7.

This episode turns on a distinction that is easy to miss. Uber could see what it was spending, but it could not say what that spending had produced. The charges accumulated until they consumed the annual budget in April. Yet president and chief operating officer Andrew Macdonald said the company still could not connect its growing use of Claude Code to the consumer features it was producing. “That link is not there yet,” he said. His statement says the link between that use and the consumer features was not yet available to senior leadership.[6]

Uber’s experience makes the problem from Chapter 1 concrete. The company was buying a resource priced by use while budgeting for a cost it expected to remain fixed per person. Public reporting does not establish whether Uber made the category error described in Chapter 1 or understood the cost model but failed to carry that understanding into its forecast. It does show that nothing in the public account identifies a control that changed the course of the spending, or a measure that connected the use to a return.

Public reporting does not reveal enough to determine where responsibility lies between Uber and its provider. The public account does not establish the contract’s specific terms, what consumption data Uber could see or when it became available, or whether the provider offered spending limits or alerts that the company declined. That question must remain open. The argument that follows does not depend on the answer. Whatever controls a provider offers, an organization must be able to govern its own consumption.

2.1 Work moves, and so does its record

That governance begins by treating AI consumption as a flow rather than a fixed annual total. Organizations already manage flows throughout their operations. Orders move from customers through warehouses

6. “Uber burned through its entire 2026 AI budget in four months,” *Fortune*, May 26, 2026, accessed August 30, 2026. Reported executive commentary on whether the spending was justified.

to their doorsteps. Invoices arrive from suppliers, pass through accounts payable, and result in payments from the company's bank account. Managers track how quickly each flow moves, where it is going, and where it can stall or leak.

AI consumption behaves the same way. Chapter 1 defined its basic unit: the consumption event, a single use of an AI system that consumes metered computing resources. One event is a transaction. Thousands of events across many people and systems form a stream whose volume and cost change with the work being done. That stream requires a different kind of management.

Transactions and streams require different forms of control. A transaction can be governed through approval: someone with authority reviews it and says yes or no. Approval, however, is limited by the approver's attention. It works for a purchase order. It cannot scale to 5,000 engineers, each deciding every few minutes whether to ask an AI system for help.

The distinction is practical, not merely semantic. A person with the proper authority can review and approve a single transaction. No one can do the same for every event generated by 5,000 engineers as they work. A flow must instead be managed over time: the organization decides in advance what is allowed to run, records what actually runs, and compares actual consumption with the plan.

DEFINITION

Flow

A continuous stream of related activity within a deployment, described by its rate, direction, and the records kept about it. An organization manages a flow by governing it over time rather than approving the deployment once.

2.2 Three flows, not one

An AI deployment produces three distinct flows, each handled by different parts of the organization.

The first is the work itself. Employees and systems send requests to models, which return outputs that feed into products, documents, decisions, and code. This is the activity the organization bought the tool to perform, and it is the only flow the organization is trying to expand.

The second is the record of that work. Somewhere, something writes down what happened: which team made the request, against which system, at what volume. Chapter 1 established that the provider always holds a meter, because the provider must bill and must manage its own capacity. Whether the buyer holds an equivalent record is a separate question, and the answer is frequently no.

The third is the flow of money and value. The meter connects cost to the work. The organization's use of the output connects that work to value. These two belong together because neither tells a manager enough on its own: cost without a corresponding return is an expense report, while return without its cost is a testimonial.

Figure 2.1 sets the three side by side. They are the usage flow, the record flow, and the cost-and-value flow. Naming them matters less than noticing that they are three separate things that can each be in a different condition. An organization can have a healthy usage flow and no record flow at all, which is the ordinary case in the first year of a deployment.

The distinction matters because ownership is divided across the organization. Engineering owns the usage flow because it decides what gets deployed and who receives access. The team that operates the platform owns the record flow, if such a record exists. Finance owns the cost side of the third flow as soon as an invoice arrives, while the business unit expected to benefit owns the value side. That leaves three flows, four owners, and no single person responsible for the whole system.

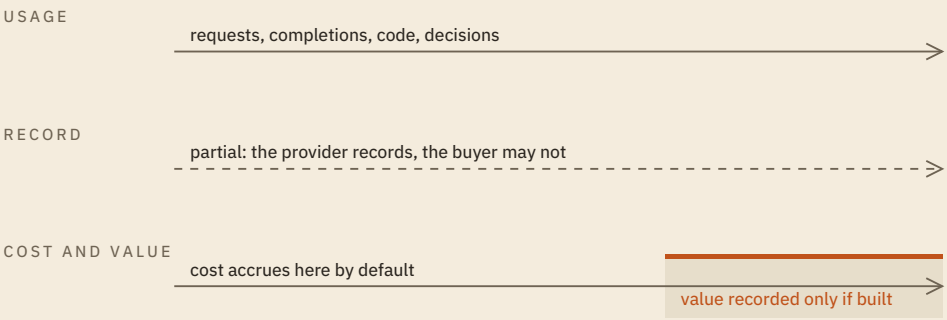

Figure 2.1 *The three flows of a deployment. Usage runs continuously once a tool is adopted, and cost runs with it. The record flow is the one that must be constructed, and the value half of the third flow is constructed too. The solid lines run whether or not anyone attends to them; the dashed line and the tinted block exist only where an organization builds them.*

A healthy usage flow has a rate the organization can measure and report. Managers can say how many requests were made last month, which teams or systems generated them, and whether usage is rising. They do not need to approve each request. They need to see the flow those requests create.

A healthy record flow has a useful grain: the level of detail needed to support a decision. A monthly total does little more than repeat the invoice and gives a manager nothing to act on. A record that shows what each team, product, and system consumed reveals where the manager can intervene. The test is simple: the record must answer a question the invoice cannot.

A record flow must be captured somewhere, and identifying that location makes it easier to manage. In practice, the records sit in one of three places. The provider may hold them in a usage console available to the customer. The organization may collect them in a gateway or platform positioned between its users and the model. Or each application may log its own calls.

Only the records kept in the gateway or applications belong to the buyer. That ownership matters when the organization changes providers, adds a second model, or needs to answer a question the provider's console cannot. Records held by the organization survive those changes. Records available only through the provider may not.

A healthy cost-and-value flow is the rarest of the three because cost and value must be measured well enough to compare, usually by people who do not work together. Finance can calculate cost on its own. Measuring value requires the business to decide before deployment which outcome the tool is expected to improve.

The three flows also move at different speeds, so they cannot be reviewed on the same schedule. Usage changes daily because it follows the work. Cost becomes clear each month when the invoice closes. Value, when measured at all, emerges over quarters because changes in how people work take time to appear in business outcomes.

Reviewing all three on the same schedule means examining one too early and another too late. In practice, managers should monitor usage continuously, reconcile cost with the plan each month, and review value on the cycle the business already uses for the relevant outcome.

2.3 Funded as a project, run as a flow

Organizations fund most AI deployments as projects. Someone writes a business case, states the expected cost and benefit, and asks for approval once. The organization then releases the money, buys the tool, and marks the project delivered when employees begin using it.

Once the tool is in use, however, the deployment operates as a flow. Consumption continues every working day and rises or falls with the work for as long as the tool remains in use. The funding decision happened once, but the spending continues. The opening case shows the mismatch clearly: the organization approved an annual budget that the deployment consumed in four months.

This mismatch creates a pattern managers will recognize. Once the organization marks the project delivered, nothing requires anyone to revisit the business case. Consumption, meanwhile, continues to change with the work. By the time anyone compares actual use with the original case, that case describes a deployment that no longer exists.

The mismatch also determines how often managers must review the deployment. A funding decision made once cannot govern spending that changes every day, yet project funding usually leads to an annual review. That schedule gives managers no chance to respond when consumption rises during the year. They must review spending often enough to catch a meaningful increase before it pushes the deployment off budget.

The same mismatch distorts how the organization measures value. A project business case states an expected benefit at the moment of approval, before anyone can test it. If no later review compares that forecast with actual results, the expected benefit becomes the organization's permanent belief about what the deployment delivers. What began as a forecast is treated as a result, even though no step in the process verified that it came true.

2.4 An unmanaged flow does not stay still

A manager reading this far might reasonably conclude that an unmanaged flow is simply an unmeasured one, and that the organization can fix the problem by measuring it later. That conclusion is wrong.

An unmanaged usage flow expands wherever the tool makes work easier. This is not a criticism of the people using it. A coding assistant that drafts code well will be used more than one that drafts code badly, which is exactly what the organization wanted. Rising usage is evidence of adoption, but it is not proof of success. Success depends on what that usage produces.

An unmanaged record flow does not simply remain incomplete. It deteriorates because the information needed to build the record becomes harder to recover over time. If a request is not attributed to a team when it is made, the organization usually cannot reconstruct that attribution six months later. The provider's invoice will show what the organization spent in March, but not which product line produced the cost. The provider cannot report information it never collected.

In an unmanaged cost-and-value flow, cost accumulates but evidence of value does not. This is the central asymmetry of the chapter. The organization receives cost figures automatically because the meter is already running and the invoice already has an owner. Evidence of value requires deliberate work. Someone must define what the deployment is supposed to improve, measure the starting point and the later result, and determine how much of the change came from the deployment. The tool does none of this work, and the invoice will never supply the answer.

The three tracks are separated to make diagnosis easier, not because the underlying flows run independently. The record flow supports the other two. Without reliable records, managers cannot compare actual usage with the plan or connect a change in business results to the AI activity that may have produced it.

DEFINITION

Cost-value asymmetry

The condition in which metering records a deployment's cost automatically, while its value is measured only when the organization has deliberately built a way to measure it. As a result, the organization may know precisely what the deployment costs without knowing what that spending produces.

Consider an engineer who spends a morning working on a difficult migration and asks an assistant for help forty times. On that day, the organization could attribute those requests to the migration project because the engineer knows what work they supported and the system records which account made them. A month later, the engineer has moved on, and the requests have been absorbed into a monthly total. A quarter later, the invoice may be the only surviving record. Nothing had to be deleted. The context linking the requests

to the project simply became impossible to recover, and later diligence cannot reliably recreate it.

2.5 Why the record flow is the one that gets skipped

Of the three flows, the record flow is the one organizations almost always build last. The delay is structural, not the result of carelessness.

A record flow produces no immediate benefit. The usage flow begins supporting work as soon as employees use the tool, while the cost flow appears automatically through billing. A record flow creates value only later, when someone asks a question that its records can answer. Until then, its benefit remains entirely in the future.

The record flow also has no obvious owner. Engineering has shipped the tool and moved on. Finance receives an invoice it can reconcile. The team that will eventually need the record is responsible for justifying the spending, but it does not yet know which questions the organization will ask. The work therefore falls between three functions, even though each has completed the task it sees as its own.

The third reason is the most important: the absence of a record remains invisible until someone needs it. A broken usage flow announces itself when employees report that the tool is down. A cost overrun announces itself when the invoice arrives. A missing record says nothing. The problem surfaces only when a senior executive asks how the tool improved the business. By then, the honest answer is that no one can say.

This is why the remedy begins with a deliberate decision. An organization does not acquire a record flow by accident. Someone must decide to build and fund it before the question it needs to answer has been asked.

2.6 All of the cost, an unknown fraction of the value

An organization running an unmanaged deployment records all of the cost but only an unknown fraction of the value. That is the result of cost-value asymmetry.

The important word in that sentence is *unknown*, not *small*. The claim is not that deployed AI produces little value, and this chapter takes no position on how much value any particular deployment creates. The claim is narrower: billing makes the cost figure complete, while the value figure remains partial because no process requires the organization to make it complete.

The two figures cannot be compared as though they meet the same standard of completeness. If the value measure omits benefits that were never recorded, comparing it with the complete cost will make the deployment appear worse than it is. Cutting the deployment on that basis would turn a failure of record keeping into a management decision. But incomplete evidence can also distort the comparison in the opposite direction. If the organization treats a few exceptional results as representative of the entire deployment, the value figure will make the deployment appear better than it is.

A hypothetical example makes the arithmetic visible. Suppose a deployment costs one million dollars over a year, a figure the organization can verify from its invoices. The organization has measured the return from two of the five workflows the tool supports, and those two show a combined benefit of four hundred thousand dollars. A manager comparing that return with the full one-million-dollar cost would see a serious loss.

Now consider what the ratio actually compares. The numerator covers two workflows, while the denominator covers all five. Both figures are accurate, but they describe different scopes, and nothing in the ratio reveals the mismatch. The three unmeasured workflows might contribute nothing, or they might contribute more than the two measured workflows. The organization has no basis for either conclusion. That is what

unknown means. Figure 2.2 shows each quantity alongside the scope it covers.

Organizations often reach for adoption figures at this point, and not out of laziness. Adoption data is readily available because the same systems that record cost often produce it automatically. It can be reported without anyone defining what the deployment was supposed to improve. It has every property a reporting metric needs except relevance to the question of value.

Adoption answers a question about the usage flow: whether people are using the tool. Value answers a question about the business: whether the tool improved an outcome. The two measures are related, but they are not substitutes. A tool can be used constantly without producing a measurable improvement, while a tool used by a few people in the right place can pay for the entire deployment.

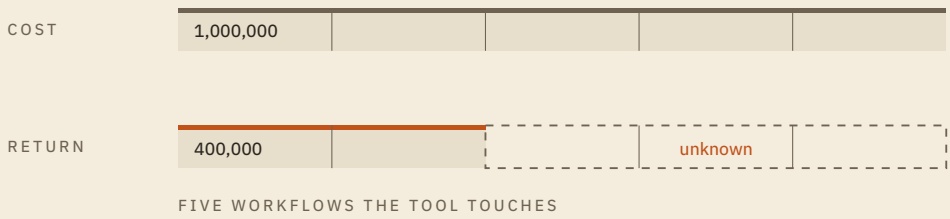

Figure 2.2 *Two quantities of different scope. The cost figure is complete and covers all five workflows, because billing covers all five. The measured return covers the two workflows somebody chose to measure. Setting one against the other produces a ratio whose numerator and denominator describe different things, and the three unmeasured workflows are not zero. They are unknown.*

The substitution is easy to spot once it has a name. Whenever a value claim rests on a number describing how much the tool was used, usage data has answered the value question. That substitution is rarely dishonest. It happens because the only complete measurement available answers the wrong question.

The disciplined response is not to guess what the other three workflows produced. It is to attach the scope to the comparison: four hundred thousand dollars of return from two measured workflows, compared with the share of the one-million-dollar cost attributable to those same workflows. That statement is longer and less quotable, but it is the only comparison a manager can act on.

Uber's position in the opening case shows this first error in practice. The company could total what it had spent, while its president and chief operating officer publicly asked what it had received in return. Both conditions could exist at once because the organization produced the two figures in different ways, not because its engineers had done anything wrong.

> **DATED: JULY 2025**
>
> The pattern is not confined to one company. A 2025 MIT NANDA study of enterprise deployments reported a striking gap: 95 percent of integrated enterprise AI pilots produced no measurable impact on profit and loss, even though a far larger share of organizations reported piloting or deploying such tools.[7] The headline drew substantial methodological criticism after it circulated, and published accounts of the study report different interview counts.[8]

The methodological dispute does not need to be settled for the finding to illustrate the problem, because the crucial phrase is *no measurable impact*. In that statistic, a pilot that created unmeasured value is indistinguishable from one that created no value. This is cost-value asymmetry at market scale: the study could not observe what the participating organizations had not recorded.

---

7. Aditya Challapally, Chris Pease, Ramesh Raskar, and Pradyumna Chari, "The GenAI Divide: State of AI in Business 2025," *MIT NANDA*, July 2025. Reported headline finding and adoption figures.

8. Aditya Challapally, Chris Pease, Ramesh Raskar, and Pradyumna Chari, "The GenAI Divide: State of AI in Business 2025," *MIT NANDA*, July 2025. Criticism and reported methodological variation, stated once here.

2.7 Why this requires governance rather than attention

The registry states the consequence formally. Once a deployment reaches scale, economic control requires a governing apparatus. Diligence from the people running the deployment is no longer enough.

THEOREM 2 · THM-004

Scaled AI Deployment Requires Cost Governance for Economic Control

Within a defined deployment and cost boundary, if:

(i) AI activity consumes resources and therefore accrues cost as it is used;

(ii) deployment has scaled beyond isolated or occasional use;

(iii) the resulting consumption varies with work rather than with headcount; and

(iv) no apparatus records, attributes, and constrains that consumption;

then the organization cannot exercise economic control over the deployment, whatever the diligence of the people operating it.

The final clause carries the practical meaning. Economic control does not come from careful individual behavior. Five thousand engineers can each make a reasonable decision and still produce a level of total consumption that none of them intended. Each decision may be defensible on its own, but no engineer can see all the others. Governance provides the aggregate view that none of the individual participants has.

A governing apparatus for a deployment must contain three elements. Without all three, the organization does not have economic control.

The first element is a record: a system that records what the deployment consumed at a level of detail finer than the invoice. The second is attribution: a rule that assigns each unit of consumption to a team,

product, or purpose. A record without attribution tells the organization how much it spent but not who or what produced the cost. The third is a constraint: a limit, budget, alert, or approval that can stop or slow consumption when it departs from the plan.

These elements are separable. An organization can have a record without attribution, or a record and attribution without a constraint. A dashboard that no one acts on is a record flow with no constraint attached. It may reveal the problem earlier, but it cannot change the activity producing it. Visibility is an improvement. It is not control.

One question follows immediately, although this chapter will not answer it: who owns the apparatus. Chapter 1 described the problem as a category error, and Chapter 3 names the discipline that resolves it. For now, the important point is that ownership does not fall automatically to whichever function noticed the problem first. Finance noticing an overrun does not make cost governance a finance-only system, just as engineering noticing latency does not make reliability an engineering-only concern.

This distinction also explains why the remedy is never simply to spend less. An organization that cuts consumption without building the record flow lowers its cost but learns nothing. It remains in the same unmanaged position at a lower volume. Chapter 8 builds the record flow, and Chapter 10 turns that record into a budget. The order matters: the record must come before the budget because, without a record, no one can compare actual consumption with the amount planned.

2.8 What this chapter does not claim

Before turning to the diagnostic, the chapter must establish four limits. Each prevents a conclusion that a reader might reasonably draw but that the argument does not support.

First, the chapter does not claim that the three flows are a scientific finding. The taxonomy is this book's organizing framework, chosen because it makes the condition of a deployment visible on a single page. THM-004 is the formal registry result. As Chapter 3 explains, the theorem formalizes the economic consequence of the third flow's asymmetry; it does not establish the taxonomy itself.

Second, the chapter does not claim that unmanaged deployments are wasteful. Waste describes the relationship between value and cost, and the argument here is that the value side of that relationship remains unmeasured. Calling a deployment wasteful from its cost figure alone is one error. Calling it successful from its adoption figure alone is the same error in the opposite direction.

Third, the chapter does not claim that measurement is free. Chapters 8 and 12 treat record keeping and value measurement as investments with returns of their own, not as routine hygiene. Building a record flow requires engineering time. Building a value measure usually requires even more work, because the organization must decide what to measure before it knows the result.

Fourth, the chapter does not claim that every deployment needs all three flows managed to the same standard. A pilot with twenty users and a fixed monthly cost does not need a detailed attribution system. Scale is the second antecedent of THM-004 precisely because the theorem does not apply below it. Managers must judge when a deployment has crossed from a small pilot into scaled use, and no single event announces that crossing when it occurs.

CRAFT SECTION

Mapping the three flows

The **three-flow mapping** is the diagnostic used throughout the rest of the book. It returns in Chapter 8, which builds the record flow that organizations are structurally most likely to skip. The mapping takes one named deployment and determines the condition of each flow in order,

using evidence that supports each diagnosis. It is deliberately short enough to run on a live deployment in an afternoon rather than staff as a separate project.

Two practices determine whether the mapping changes a decision. First, name the evidence behind every diagnosis. Without evidence, a diagnosis cannot be challenged or corrected. Second, map a deployment that is already live rather than one still being planned. A planned deployment has no flows yet, only intentions about what those flows will become.

Step 1. Name the deployment and define its boundary. Identify the tool, the user population, and the period under review. If the boundary is not explicit, it will shift while the diagnosis is being made.

Step 2. Trace the usage flow. Identify who or what sends requests, the rate at which requests occur, and whether that rate is rising. The evidence must be a count over time, not an impression. If no count exists, record two findings: the record flow lacks the count, and the usage flow cannot be considered managed without it.

Step 3. Trace the record flow. State what the organization records, who records it, and at what level of detail. One question settles the diagnosis: can the organization determine which team consumed what without asking the provider? An invoice total alone is not a record flow.

Step 4. Trace the two halves of the cost-and-value flow. On the cost side, identify who owns the invoice and which teams, products, or activities the charges can be assigned to. On the value side, state what the deployment was supposed to improve, whether the organization measured that outcome before deployment, and whether it has measured the outcome since.

Step 5. Diagnose each flow as managed, partly managed, or unmanaged, and state what evidence would change the diagnosis. A diagnosis that no evidence could overturn is an opinion.

Two tests distinguish the three states, and a flow must pass both to be considered managed. First, the organization must be able to state the condition of the flow from a record it holds itself. Second, that record must be detailed enough to support a decision. A flow is partly managed when it passes only one test or when the organization can state only part of its condition. It is unmanaged when the organization cannot state the condition at all. An organization that can attribute only half of its consumption therefore has a partly managed record flow. The cost-and-value flow receives a separate diagnosis for each half because different people build those halves to different standards.

Two errors follow from these distinctions. The first is answering step 3 with a number supplied by the provider. A provider's usage console is evidence of the provider's record flow, not the buyer's. If the organization can see consumption only by logging into a vendor portal, it remains dependent on the provider for its own record. The second error is answering the value half of step 4 with an adoption figure. The share of employees using a tool describes the usage flow. It becomes evidence of value only when the organization has established that use produces the outcome the deployment was funded to improve.

Applied to Uber on the evidence publicly reported, the mapping produces a harsher diagnosis than a casual reading might suggest. The usage flow is partly managed. The direction of adoption was visible from month to month, but adoption counts people rather than requests. Section 2.2 requires the usage flow to show how many requests were made and whether that number was rising. The public account does not establish that anyone could do so.

The record flow cannot be settled from the public account, and the reason is the first of the two tests. A per-engineer cost range was reported, which shows the figure existed somewhere. It does not show that Uber held it, because the provider always keeps a record and the buyer holds one only if it built one. Nothing reported shows that anyone attributed the consumption to teams or compared it with the plan. The cost half of

the cost-and-value flow is complete, while the value half appears unbuilt, as the chief operating officer's statement indicates.

The mapping therefore does not identify a fully managed flow anywhere in the deployment, even though a casual reading might treat the adoption evidence as proof of success. That is the diagnostic doing its job. An adoption percentage is often the most readily available number in a deployment, but it measures people rather than consumption.

A second mapping shows what a different result looks like. The deployment below is constructed for this exercise and does not come from a published source. No cited deployment was available for which the mapping produced three fully managed flows. Stating that limitation is better than presenting a constructed case as though it were real.

A retailer deploys an AI assistant to two hundred customer-service agents. The contact-center platform records usage volume by agent and shift because it already records other activity at the same level of detail. The provider bills the service organization monthly and breaks the cost down by queue. Before deployment, the retailer measured average handling time and first-contact resolution for six months. It continues to measure both outcomes.

The usage flow is managed for an ordinary reason: the usage record is a by-product of a system the retailer already operates. The record flow is managed at a useful level of detail because the retailer can attribute consumption to a queue. Both halves of the cost-and-value flow are also managed. The retailer can assign costs by queue, and it measured the intended outcomes before deployment and continued measuring them afterward.

The difference was not better intent. The retailer already had a measurement system that fit the deployment, and the deployment was small enough to make that fit inexpensive. The work required to manage the flows is easiest before a deployment scales. This creates an awkward tension: the apparatus is easiest to build at the point when it is hardest to justify.

The mapping is deliberately strict about partial answers because recording an answer as partial makes the next conversation possible. Its purpose is to produce a short list of what the organization does not know, so someone can decide which missing facts are worth the cost of obtaining.

In a first-year deployment, the mapping can find a healthy usage flow, an absent record flow, and a cost-and-value flow with only its cost half built. The missing record surprises where the missing value measure does not.

Chapter summary

A deployment produces three flows: the work itself, the record of that work, and the cost and value attached to it. Each flow can be in a different condition, and different parts of the organization may manage them.

An unmanaged flow does not remain still. Usage expands wherever the tool makes work easier, the information needed to build records becomes harder to recover, and cost accumulates automatically.

Billing the costs will happen automatically. Value is measured only when the organization deliberately builds a way to measure it. An unmanaged deployment therefore gives the organization a complete cost figure but evidence of only an unknown fraction of the value. Unknown does not mean small.

Organizations fund deployments as projects but operate them as flows. The funding decision happens once, while the consumption it authorizes continues daily and changes with the work. Without recurring review, the business case quietly becomes the organization's permanent belief about what the deployment delivers. A forecast is treated as a result.

Organizations skip the record flow most often for structural reasons. Its benefit appears only later, no single function clearly owns it, and its absence remains invisible until someone asks a question that requires it. The organization must decide to build the record before that question arrives.

Because consumption varies with work rather than with headcount, economic control at scale requires a governing apparatus, not merely diligent people. That is the result formalized in THM-004, and it is why the chapters that follow build the record before they build the budget.

Key terms

Flow

A continuous stream of related activity within a deployment, described by its rate, direction, and the records kept about it. An organization manages a flow by governing it over time rather than approving the deployment once.

Usage flow

The stream of consumption events a deployment generates, including requests made, work performed, and outputs returned. It is the only one of the three flows that an organization normally seeks to increase.

Record flow

The stream of information an organization keeps about its own usage, including which team or product consumed what, at what volume, and when. The provider always keeps a record because it must produce a bill. The buyer has a record only if it deliberately builds one.

Cost-and-value flow

The stream of money spent on a deployment together with the business return attributed to that spending. The two halves form one flow because neither can support an economic decision on its own.

Cost-value asymmetry

The condition in which metering records a deployment's cost automatically, while its value is measured only when the organization has deliberately built a way to measure it. As a result, the organization may know precisely what the deployment costs without knowing what that spending produces.

Three-flow mapping

The diagnostic that establishes the condition of the usage, record, and cost-and-value flows in a named deployment. The mapping also identifies the evidence that would change each diagnosis.

Discussion questions and problems

DISCUSSION QUESTIONS

1. An executive says the organization's AI deployment is well managed because the monthly invoice is reviewed and approved every month. Using the three flows, explain what that review does and does not establish.

2. The chapter argues that an unmanaged record flow decays rather than merely remaining empty. It illustrates the problem with an engineer whose forty requests lose their project context within a month. Using that case, identify what the organization would have needed to record when the work occurred and explain why no one involved had a clear reason to record it.

3. Why does the chapter insist that the crucial word is "unknown" rather than "small"? What would a manager do differently under each interpretation?

4. Chapter 1 argued that a flat price relocates the meter to the provider rather than abolishing it. Using that result, explain why a flat-rate contract does not remove an organization's need for a record flow.

5. The chapter argues that organizations skip the record flow because its benefit appears only later and no single function clearly owns it. Which function in your organization would you make responsible for the record flow, and what would that function need in order to accept ownership?

PROBLEMS

P1 · WORKED

The review that would have changed the outcome

The craft section maps the three flows in the opening case and finds a partly managed usage flow, a record flow the public account cannot settle, and a cost-and-value flow with only its cost half built. Start from that diagnosis rather than repeating it. State what the organization would have needed for a February review to change the outcome. For each flow, name the required record, its level of detail, and the person who would have needed to review it.

Begin with what the diagnosis already establishes, and do not supply a remedy that the evidence cannot support. The mapping found a usage flow measured in people rather than requests, a record flow the public account cannot settle, and no public evidence of a built value half. A February review needed one item from each flow, but those three items were not equally obtainable.

From the usage flow, the review needed a count of requests over time rather than a count of people with access. The appropriate interval was one month because the invoice closed monthly and managers needed to compare actual consumption with the plan. The adoption figures already available would not have worked. A majority of engineers can use a tool at almost any level of consumption, which is why Section 2.2 distinguishes a count of people from a count of requests.

From the record flow, the review needed consumption attributed to a team and compared with that team's share of the plan. Attribution makes a rising total actionable. The total tells a manager that something has changed, but not where to respond. The buyer must hold the record because a review scheduled for a fixed date cannot depend on asking the provider for information.

The cost-and-value flow provides the most important answer and the hardest one to assemble. The cost half was already complete because the invoices arrived and accumulated. The value half could not have been assembled in February. Measuring return after deployment requires a baseline from before deployment, and the public account does not establish that such a baseline existed. A February review therefore could have caught the rising spending but could not have established the return.

The review also needed an owner, which is the requirement most readers leave out. The three records cut across the four owners identified in Section 2.2, so someone had to bring those functions together and call the meeting. Two of the three information gaps could have been closed in February. The missing baseline could not, and that is the gap behind the chief operating officer's statement.

P2 · GUIDED

The seat forecast that did not hold

A company forecasts the cost of an AI assistant by multiplying a fixed monthly amount per employee by the number of employees expected to have access. It holds that forecast flat for twelve months. At the end of the year, total spending is three times the budget even though the number of employees with access matches the forecast exactly.

State which of the four antecedents of THM-004 this situation satisfies, and identify the specific assumption in the forecasting method that failed. Then state what the company would need to record, at what level, to produce a forecast that could have held.

Work in that order rather than beginning with the remedy. The antecedents establish whether the theorem applies. A deployment that has not scaled falls outside its scope and does not require the apparatus the theorem describes. Once the theorem applies, the forecasting question becomes relevant, and the failed assumption determines the level of detail the missing record must provide.

P3 · INDEPENDENT

Map a deployment from a published evaluation

Between January and June 2024, the Australian Government ran a whole-of-government trial of Microsoft 365 Copilot. Around sixty agencies participated. Each agency nominated or approved the employees who would receive licences, and the government issued several thousand licences. The product was priced per seat rather than by unit of use.[9] The Digital Transformation Agency commissioned the evaluation, delivered it jointly with an external firm, and

9. Microsoft, "Expanding Copilot for Microsoft 365 to businesses of all sizes," *Microsoft 365 Blog*, January 15, 2024. Per-seat pricing for the product, from the vendor's own announcement.

published the full report. Participants reported using the tool a few times a week or less and saving time when they used it. The evaluation named five limits in its method, one of which is that participants assessed the effects themselves, which could understate or overstate them. Its published methodology reports no objective measure of task time or output from before the trial.[10]

Do four things.

First, mark up the three tracks of Figure 2.1 for this trial: for each flow, show where it is instrumented and where it is not.

Second, diagnose each flow as managed, partly managed, or unmanaged, and state the evidence that settles each diagnosis.

Third, state for each flow what evidence would change your diagnosis.

Fourth, sort the three diagnoses into two groups: those based on evidence the published account establishes and those based on matters it does not address. The second group is weaker. Separating the two is the difference between diagnosing the deployment and merely complaining about the available sources.

This trial is not the opening case under another name. The organization commissioned the evaluation, published the full report, and allowed it to state the limits of its own method. Determine what that evaluation made knowable and what it could not establish.

P4 · INDEPENDENT

The business case that became a belief

An organization approved an AI deployment two years ago based on a business case that projected a 20 percent reduction in the time required to complete a routine task. The deployment is still running. No one has revisited the projection, yet the current year's planning documents cite it as the deployment's contribution.

Identify how the status of the 20 percent figure changed between the business case and the planning document, and name the step in the organization's process that should have prevented that change. Then state the smallest adjustment to the review schedule that would keep it from happening again, and

10. Digital Transformation Agency and Nous Group, "Evaluation of the whole-of-government trial of Microsoft 365 Copilot," *Australian Government Digital Transformation Agency*, 2024. Trial period, participating agencies, licence allocation, and the evaluation's statements about its own method.

explain why changing the business-case template alone would not solve the problem.

Interleaving: question 4 and problem P2 require Chapter 1's results. P2 uses the resource consumption model and question 4 uses meter relocation.