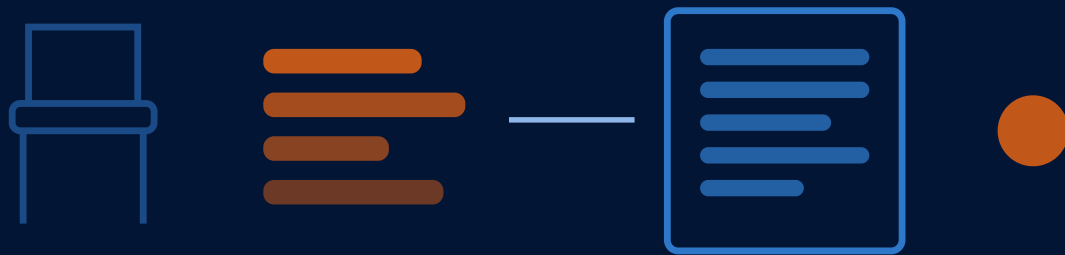


The User Who Left the Room

What a trained person was silently supplying, and who supplies it now



For twenty years a user story worked because a person stood at the other end of it, completing the specification at runtime, for free. Remove that person and the judgment does not leave with them. It becomes context nobody wrote down, in a system that acts on its own.

Yoram Friedman, MD

Physician and enterprise product leader · SAP, Walmart, Microsoft, regulated health technology

Drawn from the Agentic AI for Product Leaders series · agenticaiprductmanagement.com

THE ARGUMENT

She has not written code. She has written policy.

It is late, and an engineer is turning a goal into an agent. The ticket she picked up reads the way tickets have read for twenty years.

As a support agent, I want to see the customer's complaint history so I can resolve the complaint fairly.

She has shipped a hundred features from stories like it, and she starts this one the same way. Then she notices what is missing. She is not building a screen for a support agent to read. She is building the thing that resolves the complaint, start to finish, with no support agent in the chair.

So she opens the model and begins writing the parts the story never mentioned. What the refund ceiling is. Which records the agent may change and which it may only read. When a payment needs a human signature. What has to be kept as proof once the case is closed.

None of it was in the story, because the story described a person who would have known all of it without being told. By the time she ships, she has made a dozen decisions that were never hers to make.

The specification was never complete. It was completed at runtime, for free, by the person in the chair. Take that person out and the judgment does not leave with them. It becomes context nobody wrote down, sitting in the one place it cannot stay: unstated, in a system that acts on its own.

PART ONE

The half of the specification nobody was paying for

Take the one line the engineer started from. Written for a person, it needs nothing more.

The support agent already knows that store credit under fifty dollars is hers to grant, that a refund over two hundred goes to a supervisor, that a customer who has charged back twice gets escalated rather than refunded, and that the note explaining the decision has to be saved for the auditors.

None of that knowledge was free, and none of it was in the story.

WHAT THE STORY SAID

As a support agent, I want to see the customer's complaint history, so I can resolve the complaint fairly.

WHAT THE READER SUPPLIED, FREE, AT RUNTIME

Store credit under fifty dollars is hers to grant

Refund over two hundred goes to a supervisor

Two chargebacks means escalate, not refund

The note explaining it is kept for the auditors

None of it in the story. None of it free to the company.

WHAT IT COST TO PUT IT THERE

Interviewed for judgment, background, training

Company onboarding

Support-agent training

Policies read, and signed for

Supervised probation beside a senior agent

Named accountability for the outcome

The agent sat no interview.

It completed no onboarding, signed no policy, and served no supervised probation.

The story did not get more complicated.

The reader changed, and the reader had been carrying half the specification.

Figure 1. The story on the left is the artifact the company paid for. The column beside it is what the reader supplied at no marginal cost, and the column on the right is what it took to put it there. One of those three things is now missing, and it is not the story.

She was interviewed for her judgment, her background and her training before she was hired. She went through company onboarding and then support-agent training. She was handed the relevant policies, asked to read them, and made to sign that she had. Then she worked beside a senior agent as a trainee until someone decided she could be trusted on her own.

The user story could stay silent on the refund ceiling because the company had spent months and an entire hiring pipeline making sure the person who would read it already knew the ceiling by heart.

Written for an agent, all of it has to be on the page before the agent runs: the ceiling, the escalation trigger, the chargeback rule, the record that has to survive the case. The agent sat no interview. It completed no onboarding. It signed no policy and served no supervised probation.

The story did not get more complicated. The reader changed, and the reader had been carrying half the specification.

You are not building a better tool

Notice what the story was specifying all along. When its target was a person, it described a tool: a way to make someone who already knew the job faster at it. A dashboard is a tool. A queue is a tool. A history view is a tool.

Take the person out of the room and the tool is beside the point. What you have to build now is the thing the story never described, because it never had to. Not a sharper instrument for a

competent human. The competent actor itself.

That is a different product, not a harder version of the same one, and most of the difficulty teams run into is the cost of not having noticed the substitution.

PART TWO

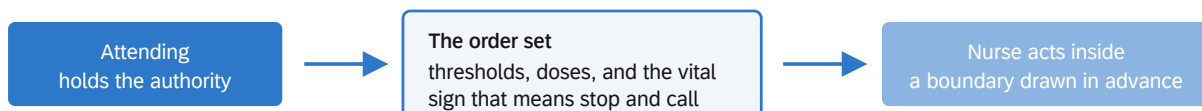
Medicine settled a version of this a long time ago

Worth borrowing, because the stakes there never allowed the shortcut.

An attending physician cannot stand at every bedside at every hour. So the judgment is not handed to the night nurse as a description of what the attending would do if she were there. It is handed over as an order set and a protocol: the thresholds, the doses, the vital sign that means stop and call.

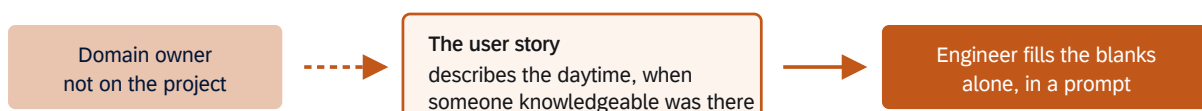
The nurse acts inside a boundary the attending drew in advance, because the attending holds the authority and the nurse, at three in the morning, needs the decision already made rather than improvised.

HOW MEDICINE HANDS OVER JUDGMENT



The decision is already made, because at three in the morning it cannot be improvised.

HOW MOST TEAMS HAND OVER JUDGMENT



The authority and the decision have come apart. The person holding the ticket is not the person who holds the judgment.

Figure 2. Two ways of handing judgment to something that will act without you. In one, the person who holds the authority writes the boundary down in advance and the actor works inside it. In the other, the person who holds the authority is not on the project, and the blank gets filled by whoever is holding the ticket.

An agent needs exactly what the night nurse gets, and it needs the people who hold the judgment to be the ones who write it down. What it usually gets instead is a story that still describes the daytime, when someone knowledgeable was always in the room, handed to the last engineer to touch it.

The order set is worth studying for a second reason. It is not a description of good practice, and it is not a training document. It is executable at three in the morning by someone who is

competent, tired, and alone, and it survives contact with a case nobody anticipated because it says what to do when the situation leaves the protocol. Most agent briefs fail exactly that test.

PART THREE

The deliverable that did not change

Look at what has moved on the team, and what has not.

Engineering has already changed its unit of work. A strong team building an agent does not hand over a sequence of screens to implement. It works from a statement of intent and the limits around it, an outcome and the boundaries on how the agent may reach it. Watch a capable engineer build an agent and you will see her writing conditions and constraints, not procedures. The unit moved from the steps to the goal and the guardrails.

Design is moving too. The designer's object is no longer only the screen the user reads on the way to a decision. It is the moment a human is pulled back into the loop to approve or override, and what that human sees when they arrive: enough to make a real judgment, or a summary polished enough to rubber-stamp. When the user is doing the task, you design the task. When the agent is doing the task and a person supervises it, you design the interruption. Those are different crafts.

The product manager's deliverable, in most organizations, has not moved at all. It is still a user story and a set of screens describing what a person would do, in a product where a person increasingly does not do it.

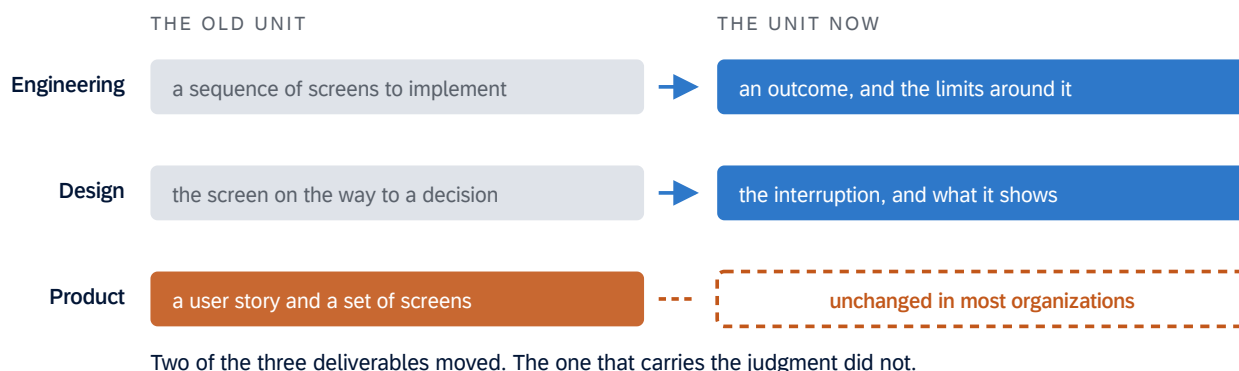


Figure 3. Two of the three deliverables changed shape as soon as the software started acting. The third did not, and it is the one that was carrying the judgment.

Where the decision actually lands

Because the story reaches the engineer carrying none of the conditions the running system needs, the engineer supplies them. Alone, in a prompt, at the one altitude in the whole organization least equipped to own them.

She is not the person who has carried the refund rules in her head for fifteen years. She is not the one accountable when a payment goes out that should not have. She is the one holding the ticket at the moment the blank had to be filled.

A product manager's deliverable exists to make the next person's work possible. When it still describes a user who will not be in the room, it makes the designer guess at the approval moment and the engineer guess at the policy, and it quietly reassigns decisions that belong to the business onto whoever is downstream when the deadline hits.

PART FOUR

The audit that finds the missing half

There is a mechanical way to find what the departed user was carrying, and it takes an afternoon rather than a project.

Take one story from a product you are about to make agentic. Name the person who was going to read it. Then write down what that person knew when they sat down, in four categories.

ONE · AUTHORITY

What were they allowed to decide alone, and where did their authority stop? A number, not a principle. Fifty dollars, two hundred dollars, this record but not that one. If the organization has never written the number down, that is the finding, and it is a finding about the organization rather than about the agent.

TWO · EXCEPTIONS

What made them stop and hand the case to someone else? The chargeback pattern, the account that smells wrong, the request that is technically outside policy but right on the merits. This is the category people find hardest to articulate, because it lived as recognition rather than as a rule, and it is the category the agent will meet on day one.

THREE · EVIDENCE

What did they write down, for whom, and how long did it need to survive? A note for the auditor is a different artifact from a note for the next agent on the case, and the person in the chair knew which one they were writing.

FOUR · ACCOUNTABILITY

Who answered when they got it wrong, and what happened next? If the honest answer for the agent version is nobody, that is not a governance gap to close later. It is the design problem, and it is yours.

Then compare that list against your actual specification. Every line that is in the first and not the second is currently being supplied by an engineer in a prompt, at deadline, by inference.

The test that ends the argument

Hand your brief to someone who knows the domain and did not write it, and ask them to name the case where the agent would get it wrong. If they can do it in under a minute, the brief is describing the daytime. Every experienced practitioner carries a catalogue of the cases that look ordinary and are not, and none of that catalogue is in a user story, because the story was written for someone who already had it.

PART FIVE

Who actually holds the judgment

The hardest part of this work is not writing the judgment down. It is that the person who holds it is usually not in the room where the writing happens.

The support agent with fifteen years of refund decisions is not on the project. The claims adjuster who knows which exclusions are contestable is not in the sprint. The nurse who knows which vital sign is the one that matters at three in the morning is on shift. They are the source, and the org chart has them nowhere near the specification.

That is the actual work, and it is product work rather than engineering work. Finding the person who holds the judgment, getting them to say what they know, and turning what they say into something a system can act on before it acts. Not a description of what a good practitioner would do. A boundary a system can be held to.

There is an uncomfortable corollary. Some of what that person knows is wrong, or out of date, or is a habit that was never policy. Writing it down surfaces that, which is why the exercise is often resisted and why it is worth doing anyway. An organization that cannot state its own refund rule does not have a rule. It has a person, and it is about to replace them.

Every agentic product is an audit of what your organization actually decided, conducted by a system that will act on whatever it finds. Most organizations discover in that audit that a surprising amount of their policy was a person.

THE ONE LINE

The user has left the room. What they knew has to be put on the page.

Not as a description of what a competent person would have done, which is a training document. As a boundary the system carries before it acts: the authority with a number on it, the exception that stops the run, the record that survives the case, and the name of the person who answers when it is wrong. That is the new unit of product work, and nobody downstream can produce it for you.

Sources and status. Drawn from the *Agentic AI for Product Leaders* series by Yoram Friedman, principally *Agentic AI for Busy Product Managers* (second edition), chapter 1 and the Departed User framework, together with the two briefs and the four runtime artifacts; *Why Agentic AI Products Fail* on the work-unit shift from user story to outcome specification and on the enforcement principle; and *The Agentic AI Team* on the seam between the person who decides a boundary and the person who enforces it. The support agent, the refund thresholds and the engineer at her desk are an illustrative composite. The order set and protocol comparison is drawn from clinical practice and from the author's clinical training; it is offered as an analogy for how authority is delegated in advance, not as a claim about any specific protocol. The Departed User is the author's framework.