THE PLATFORM WEEKLY
AUTOMATE THE DIFFICULT. PRESERVE THE HUMAN.
ISSUE 09
NOV 19, 2026

The Tenth Process

What if intelligence could learn the underlying structure of manufacturing work and apply that understanding to processes it has never encountered before?



A WEEKLY EDITORIAL ON INTELLIGENT MANUFACTURING

The Tenth Process

Two days of difficult questions, nine visual explorations, and a possibility far larger than any individual workflow: what if an operating platform could learn the underlying structure of work, and apply that understanding to processes it has never seen before?

The Question Behind the Nine Images

Some weeks produce a visible achievement. Others change the questions you are asking. This week did both, but I suspect the more important result will be the question that emerged from two days of intensive investigation into manufacturing processes. Kevin and I found ourselves examining the depth of work that is so often compressed into a deceptively simple diagram. The conversation was not merely about which task follows which, or how to make a workflow easier to configure. It was about what a process actually contains when we stop treating it as a tidy sequence of boxes and begin taking seriously the relationships, exceptions, responsibilities and decisions that make it work in the real world.

The exchanges were demanding because manufacturing refuses to stay inside the clean boundaries we draw for it. A process that appears straightforward from one perspective becomes something else when the supplier cannot deliver, the evidence is incomplete, a sample fails inspection or a customer changes a commitment. Some activities must stop; others can continue. Some decisions belong to a technician, others to a factory, a brand or a customer. Information may be visible to one organization and deliberately withheld from another. A date may be a working forecast, a supplier promise or a binding commercial commitment, and confusing those meanings can turn a minor operational change into a serious business problem. Every attempt to simplify the picture exposed another layer that needed to be understood before it could responsibly be hidden.

Out of those two days came nine images, developed through the back-and-forth between manufacturing experience and architectural thinking. I do not see them as nine finished software specifications, nor as nine processes that The Platform must reproduce in exactly the same way. Their value lies in something more fundamental. They forced us to make our assumptions visible. They gave us a way to examine how work begins, how it changes, how authority is exercised, how evidence travels and how consequences spread. The images became a centre of gravity for the future of Process Management because they made it possible to discuss an operating reality that words alone can easily oversimplify.

Yet the most consequential thought arrived after the diagrams themselves. If it takes two days to understand the complexity of nine scenarios, what happens when a manufacturer needs a tenth? Or a hundredth? Do we repeat the same expensive exercise every time, gathering experts around another set of boxes and arrows? Do we ask every organization to fit itself into a library of processes somebody else designed? Or could we learn enough from the structures underneath those scenarios that the next process becomes easier to understand, configure and operate? That question changes the ambition from building a powerful workflow tool to exploring a reusable form of operating intelligence.

I keep returning to the distinction because it is easy to miss. Nine images can illustrate nine situations. Nine configurations can automate nine situations. But a platform that understands why those situations behave as they do could eventually support a tenth situation without starting from nothing. It would still need the knowledge of the people who understand the work, and it would still need explicit rules and permissions. What might change is how much of the structure has to be rediscovered, how quickly the consequences can be understood and how reliably experience can be carried forward. The tenth process is therefore not another item on a roadmap. It is a test of whether the architecture can grow more capable through use.

What Different Processes Have in Common

Manufacturing processes are wonderfully inconsistent on the surface. Developing a garment sample is not the same as producing a fabric batch, approving a quotation or coordinating a shipment across two factories. The documents differ, the people differ, the timing differs and the consequences of failure differ. A mill may care about dye recipes, machine settings, lot history and test results. A product developer may care about fit, measurements, construction, revisions and customer approval. A factory may be concerned with capacity, material readiness, cutting, sewing, quality and dispatch. To insist that all of this is one identical workflow would be to ignore precisely the knowledge that makes manufacturing work.

But beneath those differences, certain questions keep returning. What event caused this work to begin? What is the current state of the thing being worked on? Which other activities depend on it? What evidence is required before it may proceed? Who is authorized to decide? Which conditions change the route? What is permitted to continue while something else is blocked? What promise has been made, to whom, and what happens when reality no longer matches that promise? These are not merely user-interface questions. They are recurring elements of a language for describing work, regardless of whether the immediate subject is a fabric roll, a prototype, a purchase order or a delivery.

That suggests a different way of designing Process Management. Instead of treating each process as a custom chain of screens, we can investigate a common operating grammar built from events, states, dependencies, conditions, evidence, authority, timing, commitments and exceptions. A specific manufacturing process would still have its own vocabulary and rules, but it would express them through a structure the platform already understands. The same underlying engine that knows an unapproved sample

cannot advance to a particular milestone could also understand that a fabric batch cannot be released without required quality evidence. The facts and authorities are different. The pattern of a governed decision is familiar.

This does not mean that every exception can be reduced to a formula, or that the subtleties of a mill's experience should be flattened into a universal template. Quite the opposite. A useful shared structure should make those differences explicit and preserve them. One organization may require a formal approval where another uses a delegated role. A brand may allow preparatory work to continue during a sample review, while a factory's own procedures impose a stricter hold. Some evidence can be shared across organizational boundaries; other evidence is commercially sensitive. Reusability should come from a common way of representing these differences, not from pretending they do not exist.

There is a useful historical parallel in the way software evolved from individual programs toward reusable operating systems and platforms. The breakthrough was not that every application became identical. It was that certain difficult capabilities no longer had to be reinvented for every application. We are exploring whether a similar principle can apply to manufacturing operations: a foundation that understands the mechanics of work, while allowing the work itself to remain richly specific. That is a much larger proposition than adding a process designer to an ERP. It is an attempt to give an operating environment a coherent understanding of the relationships that make business possible.

Where Algorithms End and Learning Begins

The word *intelligence* is used so casually today that it can obscure more than it explains. A process that automatically moves from one step to another is not necessarily intelligent. A system that applies a rule correctly is valuable, but it has not learned anything simply by executing that rule. An AI model that produces an elegant explanation is not necessarily reliable, especially if it cannot identify the evidence and authority behind its conclusions. The two-day investigation makes these distinctions more important, because our ambition is not merely to decorate workflow automation with an AI assistant. It is to discover how algorithms and learning could work together to support an expanding universe of processes.

Algorithms provide a disciplined foundation. A dependency graph can represent which activities rely on others. A state model can distinguish preparation from review, approval from release and forecast from commitment. A scheduling algorithm can calculate the effect of a delayed activity while leaving unrelated work untouched. A rules engine can determine that a particular decision requires specific evidence or an authorized person. A permission model can ensure that a factory sees what it needs to act without exposing a brand's internal commercial calculations. None of these capabilities needs a language model to invent the answer. Their strength is that, given defined rules and trustworthy facts, they can produce results that are repeatable, inspectable and testable.

Intelligent capabilities become useful at a different boundary, where the world is less neatly structured. A technical pack may contain drawings, measurements and comments that need to be interpreted before they can become dependable records. A conversation may reveal that a supplier is concerned about a delivery date without formally changing the commitment. An unusual quality failure may resemble an earlier event that was documented in another format. A new organization may describe its process in language that does not match the terminology of an existing template. In these situations, AI could help extract meaning, identify possible relationships, surface missing information and propose a configuration or investigation. It should not silently promote an interpretation into an approved operational fact.

Learning introduces a third requirement: memory of what actually happened. If the platform observes that certain exceptions repeatedly lead to a delay, that observation may help it recognize risk earlier next time. If a proposed response is accepted by an authorized person and subsequently proves effective, that outcome may strengthen the recommendation in similar circumstances. If a recommendation is rejected because an experienced operator knows something the system missed, that correction is valuable evidence too. Over time, the platform could build a more useful understanding of patterns, provided it records the original situation, the proposed action, the human decision and the eventual result. Without that feedback, we are not describing a learning system. We are describing a system that repeats its assumptions with increasing confidence.

The engineering challenge is therefore not to teach a model nine diagrams and expect it to invent a reliable tenth. A small collection of images is not enough to establish general intelligence about manufacturing, and resemblance is not proof that two situations should follow the same rules. The more credible approach is to formalize the recurring structures, use algorithms to enforce their known constraints, employ AI to interpret new situations and compare them with relevant experience, and evaluate the results against real outcomes. New process configurations could be proposed from familiar patterns, but they would still require review, simulation and appropriate approval. In that sense, intelligence would expand the platform's ability to recognize and assist, while governance would protect its ability to be trusted.

Imagine the difference between a system that says, “This shipment is delayed,” and one that can explain what the delay means. It might identify which colourways are complete, which remain in production, whether a customer agreed to a single shipment, whether splitting the order is contractually permitted and which parties must authorize a change. It could calculate the likely consequences of several options, compare them with similar historical situations and prepare a recommendation. But it should not independently promise the customer a split delivery simply because that option appears efficient. The most useful intelligence is not the one that acts without hesitation. It is the one that understands the boundaries within which action becomes legitimate.

A System That Can Remember Why

There is another dimension to learning that manufacturing has been missing for a long time. We are very good at recording transactions and often surprisingly poor at preserving explanations. We know that a sample was approved, but not always which compromises were accepted. We know that a production date changed, but the discussion that established responsibility may live in an email thread. We know that a fabric passed inspection, but the particular settings that made a difficult batch successful may remain with the technician who ran the machine. When people leave, suppliers change or a product is produced again months later, much of that practical understanding must be reconstructed.

Will's work on fabric libraries, Bills of Process and recipe management illustrates the scale of the opportunity. The 31 reference processes he shared across fabric production, garment manufacturing and colour management contain substantial domain knowledge. My response to him was partly amused by the number, but the underlying point was serious: those references need not become 31 rigid pathways inside The Platform. They could become sources of manufacturing knowledge that inform one process, several concurrent processes or many organization-specific configurations. Their value is not the count of templates. It is the experience embedded in the steps, evidence, quality expectations and decisions they describe.

The same principle applies to the distinction between what a product should cost and what it actually costs. In our discussion of intelligent costing, we separated internal product development costing, factory costing and commercial negotiation. Each has different responsibilities, information and permissions. The agreed price is a shared commercial outcome, not permission for either party to inspect the other's private calculations. Over time, however, actual production evidence could help each organization understand variance, profitability and the consequences of its decisions. The intelligence would come from connecting expectations to outcomes without erasing the legitimate boundaries between the participants.

That is why a manufacturing memory cannot simply be a giant collection of documents fed into an AI model. It needs context, provenance and a record of authority. A measurement extracted from a technical pack is not equivalent to a measurement verified on a physical sample. A proposed recipe is not the same as the recipe used on a successful batch. An internal cost estimate is not a negotiated price. An expected completion date is not necessarily a customer commitment. If the platform loses these distinctions, it may become very fluent at giving the wrong advice. If it preserves them, it has a chance to learn not only what happened, but why the outcome was considered acceptable and under which conditions the lesson might apply again.

A particularly interesting possibility is that the system could eventually help organizations learn from one another without treating all their information as common property. The reusable pattern might be that a certain kind of exception requires an evidence check and a named authority, while the underlying customer

records, recipes and prices remain private. The platform could become more capable at recognizing structures while continuing to respect the boundaries that make commercial collaboration possible. That is not a trivial technical problem, and it is certainly not solved today. But it is a much more worthwhile question than whether we can place a chatbot beside another database.

Making the Invisible Visible

While we were wrestling with the structure of processes, the Lovable experiment was developing a very different kind of expression. Last week's *Minority Report* experience placed a product at the centre of a fluid, almost cinematic working environment. This week, the experiment began exploring two ways of presenting the same operating reality: Explore and Focus. The important idea is not that one is a futuristic version and the other a conventional version. It is that people might prefer different ways of seeing their work without being forced into different systems, different permissions or different sources of truth. A change of presentation should not change the underlying record or the authority to act on it.

That distinction is easy to underestimate. Most enterprise applications make the interface itself the organizing principle. People learn where information lives, which screen owns an action and how to move between the functions of the business. But if the operating model is genuinely unified, the interface can become a choice rather than a boundary. Someone exploring a product might want to see its relationships spatially, following materials, samples and commitments as connected parts of one story. Someone preparing for a busy production meeting might want a concentrated view of the situations that require attention. Both should be looking at the same facts, even when those facts are expressed in very different visual languages.

The one-screen Today experiment pushes the idea further. Instead of asking a person to inspect every activity, it explores how an operating environment might distinguish between what is blocked, what requires attention, what needs a human decision and what can continue. A simulated workflow laboratory now provides ways to examine these distinctions against different scenarios. These are experiments with local demonstration data, not evidence that the production MVP already contains a mature Scout intelligence layer or a universally adaptive process engine. Their significance is that they let us test the human experience implied by the architectural work. They ask what the system should reveal when it understands the underlying complexity well enough to stop displaying all of it at once.

Scout's role becomes clearer in that setting. I do not imagine intelligence as a separate destination that people must remember to visit. Nor do I think a chat window should become the centre of manufacturing operations. Scout should eventually be present where understanding is needed: interpreting a document, investigating a discrepancy, explaining the consequence of a changed commitment, identifying missing evidence or helping someone see the available choices. It should be able to explain what it knows, what it

does not know and why a recommendation deserves consideration. Above all, it should distinguish an observation from a decision, and a decision from an action that has actually been authorized.

The visual experiment and the process investigation are therefore two expressions of the same ambition. One explores how work might feel when information is organized around the person and the situation. The other asks what must be true beneath the surface for that experience to be dependable. The first without the second risks becoming theatre. The second without the first risks becoming another sophisticated system that demands too much of the people using it. The promise of Intelligent Manufacturing lies in bringing them together: deeper operating intelligence below, greater human clarity above.

The Discipline of Being Challenged

This week's discussion with Dmitrii Tikhonov brought a useful change of perspective. In preparation for the technical review, he made a point that deserves to be remembered: before debating an ambitious target operating model, he wanted to understand what actually exists today. He asked to see the current architecture, a concrete demonstration, how data and models are handled, where Scout sits, and the difference between completed, in-progress and planned capabilities. It is a simple demand, but an essential one. A compelling vision can become less credible when we use the language of the future to describe the present.

The November 18 conversation made those boundaries more explicit. The current build includes foundations such as users, roles, organizations, customization, components, products, samples and technical-pack import. The unified workflow system remains a critical work in progress. Orders, inventory and the intended Scout AI layer were described as not yet built in the main product. That account may be less dramatic than a sweeping demonstration of the future, but it gives the ambition a firmer footing. The Lovable exploration, the governed engineering work and the production roadmap serve different purposes, and we need to be precise about which capability lives where. Precision is not the enemy of imagination. It is what allows imagination to survive serious examination.

Dmitrii also challenged Kevin's approach to AI-assisted development. Kevin has been using a controlled process that emphasizes planning, comparison, implementation and review. Dmitrii argued that a more autonomous development pipeline could accelerate delivery. I find the tension revealing because it mirrors the very question we are asking of manufacturing. Where should automation be trusted to proceed, and where does the cost of an error justify a deliberate human checkpoint? Speed matters, particularly when a small team is attempting something ambitious. But a faster route to the wrong architecture can be expensive, and caution without progress has its own cost. The right answer is unlikely to be maximum autonomy or maximum control. It will depend on the risk, reversibility and authority involved in each decision.

His commercial questions were equally direct. If the idea is strong, why not begin selling immediately? If investors are interested, what can we demonstrate without overstating the stage? How will existing ERP customers adopt a new operating model without facing a disruptive replacement project? These are not peripheral concerns. They help determine the order in which the architecture must prove its value. A manufacturer is unlikely to buy an abstract promise that software will someday understand every process. A customer might, however, pay to remove a painful gap between a sales order, factory purchase orders and the commitments that follow. A credible first commercial journey can demonstrate the value of connected operating information while the wider architecture continues to mature.

There is no need to choose between the MVP and the vision, provided we stop treating them as competing products. The MVP must solve a specific problem for a specific customer, earn trust and create evidence of value. The broader experiments help us avoid building that solution into an architectural dead end. The nine images test whether our process model can become more general. Lovable tests how that model might eventually be experienced. Dmitrii tests whether our explanations, development approach and commercial sequence withstand scrutiny. These are different kinds of work, but they belong to one Intelligent Manufacturing initiative.

The First Customer and the Tenth Process

The temptation in any ambitious undertaking is to measure progress by how much has been imagined. We can describe a network of brands, mills, factories and suppliers; a system that learns from production history; an intelligence layer that recognizes exceptions before they become expensive; and an interface that adapts to the work rather than forcing the work to adapt to software. All of those ideas may be worth pursuing. None of them, on its own, establishes that a customer will trust the product, adopt it or pay for it. The next stage requires the discipline of choosing a valuable journey and proving that it works under real conditions.

I keep thinking about the relationship between that first commercial journey and the tenth process. At first glance they seem to pull in opposite directions. The first customer demands focus, limited scope and a practical answer to an immediate problem. The tenth process demands generality, reuse and an architecture that can adapt beyond the scenario for which it was initially built. Yet they can reinforce one another. If we build the first journey with a clear understanding of events, evidence, dependencies, commitments and authority, we create a foundation that can be tested against other situations. If we design only for the first customer's screens and shortcuts, we may have to rebuild the foundation every time the business changes.

The right balance will require restraint. We should not spend months trying to create a universal intelligence before delivering useful software. Nor should we dismiss the lessons of the two-day process investigation because they appear to reach beyond the MVP. The purpose of those lessons is to identify which underlying relationships are fundamental enough to preserve now. The platform does not need to

know every future process, but it should avoid assuming that every process is linear, that every approval belongs to one organization or that every exception requires the whole operation to stop. Those assumptions are cheap to make at the beginning and expensive to remove later.

If the architecture succeeds, the tenth process should not require the same effort as the first. The system would already know how to represent an event, a dependency, a conditional route, a required piece of evidence and a human authority. It might recognize similarities with existing patterns, help configure a new process, simulate consequences and identify what remains uncertain. People would still define the business rules, validate the exceptions and decide what risks they are willing to accept. As more real operating outcomes become available, the recommendations might improve. That is a credible direction for learning, not a claim that the learning has already been achieved.

I think this is what makes the past week more important than a collection of features or demonstrations. Two days of difficult conversation exposed the depth of manufacturing processes. Nine images helped make that complexity visible. The Lovable work began asking how people might experience a system that understands it. Will's research reminded us how much specialized knowledge must be preserved. Dmitrii's questions demanded that we distinguish a future possibility from a working capability and a working capability from a business someone will buy. Together, those developments suggest an ambition that is both more demanding and more coherent than simply building a better collection of enterprise applications.

The most interesting future may not be a factory in which machines make every decision, or a software platform that promises to replace the expertise of the people who keep production moving. It may be a manufacturing environment that remembers what people have learned, recognizes recurring patterns, understands the consequences of change and carries more of the invisible burden of coordination. That would leave people with better information, clearer choices and more room for the judgment that cannot be reduced to an algorithm. The question raised by our nine images is not whether we can automate those nine processes. It is whether we can build something that becomes more capable of understanding work each time a new process is encountered.

Automate the Difficult. Preserve the Human.

The Platform Weekly is a continuing editorial exploration of one Intelligent Manufacturing operating model. This issue distinguishes between current engineering, visual experiments and future learning capabilities; the nine original process images informed the discussion but are not reproduced here.