

Agentic AI in automotive product development: Building the future of intelligent engineering



Contents

01 Introduction

02 PDP requires an agentic approach

03 From AI use cases to AI agents

3.1 Evolving from use cases to an agentic workflow

3.2 Prioritization of agentic workflows and multiagent scenarios

3.2.1 Requirements → test cases (→ test execution → defects)

3.2.2 Systems Engineering zig-zag (requirements → architecture)

3.2.3 Spec-driven development

3.2.4 Change analysis and implementation

3.2.5 Release management

3.2.6 Continuous compliance

3.2.7 Data-Driven engineering (field → design loop)

3.2.8 Variant optimization

04 NTT DATA Systems Engineering Copilot

05 Implementing an Agentic PDP

06 Authors

1. Introduction

Developing a new vehicle isn't just complex. It's one of the most demanding engineering challenges you face today. Modern vehicles are highly sophisticated cyber-physical systems, combining tens of thousands of mechanical components with tens of millions of lines of software, all integrated through advanced electrical and electronic architectures.

Bringing these systems from concept to production typically takes 3–5 years, requires billions of dollars in investment and depends on global teams of thousands of engineers working in sync. At the same time, you're under constant pressure to move faster, meet regulatory requirements and continue innovating, without compromising quality.

To address this challenge, many organizations are exploring agentic AI: autonomous, goal-driven AI agents that support teams across the product lifecycle. In product development, this shift requires more than automation. It requires an engineering-grade AI foundation where agents are trustworthy, observable, traceable and aligned with established development processes and the right amount of Human-in-the-Loop, enabling teams to accelerate work while maintaining quality, compliance and technical control.

This paper examines the way an agentic AI approach can enhance your product development processes, from early requirements definition through design, testing and deployment. We focus on four key areas:

- Why automotive product development requires an agentic AI approach
- How to move from isolated AI use cases to multiagent systems with bounded autonomy and human oversight in engineering
- How the NTT DATA Systems Engineering Copilot (SECP) provides a blueprint for an agentic product development process
- What a practical transformation roadmap looks like for adopting agentic AI

“Bringing these systems from concept to production typically takes 3–5 years, requires billions of dollars in investment and depends on global teams of thousands of engineers working in sync.”

2. Product development process requires an agentic approach



Product development in automotive is uniquely complex, and leveraging AI effectively calls for an agentic approach rather than ad hoc point solutions. This section describes the automotive PDP and its challenges, then explains why the PDP's complexity aligns with key indicators for an agent-based AI strategy.

Agentic AI brings the ability to coordinate tasks, adapt to change, connect data and support decision-making across the full lifecycle. This makes it a natural fit to address both the current challenges faced by automotive organizations and the inherent complexity of the product development process.

Key challenges and how agentic AI addresses them

Speed and efficiency pressure

Development cycles need to be significantly shorter while maintaining high engineering efficiency. Traditional timelines are no longer aligned with fast evolving market demands and emerging technologies.

You need to accelerate development by running activities in parallel, reducing manual work and enabling faster decision-making without compromising quality.

How agentic AI helps

Agentic AI supports your engineering processes by generating design alternatives, preparing decision options and coordinating activities across teams. When a requirement changes, agents can update models, notify stakeholders and trigger validation activities, reducing rework and shortening development cycles.

Compliance and quality

You must ensure full compliance with safety, regulatory and process standards while managing increasing complexity. Maintaining traceability, accuracy and consistency across requirements, design, testing and documentation becomes more difficult as systems grow in scale.

How agentic AI helps

Agentic AI enables continuous compliance by monitoring regulatory changes, translating them into engineering context and identifying impacted components. This allows you to move from reactive compliance to proactive risk management while keeping development aligned with regulations.

Complexity and interdependence

Thousands of activities run in parallel across subsystems, with strong dependencies that must converge at the right time. Managing alignment across workstreams becomes increasingly difficult, especially as system complexity grows.

How agentic AI helps

Agentic AI coordinates complex environments by breaking down work into manageable tasks and orchestrating them across domains. Multiple specialized agents operate under a unified structure, ensuring alignment across requirements, design and validation activities.

Innovation and feature complexity

You are expected to deliver more advanced features, including connectivity and intelligent systems, while maintaining timelines and cost targets. Balancing innovation, complexity and efficiency requires stronger alignment across engineering domains and faster system-level decision-making.

How agentic AI helps

With model-based systems engineering (MBSE) supported by AI, your system models become active decision tools. AI helps identify inconsistencies, detect missing interfaces and assess the impact of changes across architectures. This improves collaboration and enables confident decision-making at scale.

Uncertainty and continuous adaptation

Early development phases are characterized by evolving requirements, technical uncertainty and continuous learning. You need to adapt plans dynamically while maintaining progress toward defined goals.

How agentic AI helps

Agentic AI analyzes historical data, current project status and external knowledge to recommend next steps, identify risks and support decision-making. This enables a more responsive and resilient development process.

A stylized illustration of a factory floor. In the foreground, several yellow sports cars are parked in a row. Behind them, a series of yellow robotic arms or assembly stations are visible, each with a screen and various mechanical components. The background shows a complex industrial structure with metal beams, pipes, and scaffolding, all rendered in a dark, industrial aesthetic with a blue and yellow color scheme.

“Multiple specialized agents operate under a unified structure, ensuring alignment across requirements, design and validation activities.”



Knowledge management

Valuable knowledge exists across data, projects and teams, but it is difficult to capture, connect and reuse effectively. Limited access to structured knowledge, combined with talent shortages and workforce transitions, reduces your ability to learn from experience.

How agentic AI helps

Agentic AI connects data, decisions and outcomes across the lifecycle by maintaining traceability between requirements, design, code and validation. It enables faster impact analysis, supports audits and continuously learns from new data to improve future decisions and processes.

Autonomy and human collaboration

Many engineering activities can be automated, but human expertise remains critical for decision-making and oversight. You need to balance automation with control, ensuring that engineers remain central to critical decisions.

How agentic AI helps

Agentic AI works alongside your teams, automating routine tasks while providing insights and recommendations. It explains its outputs and learns from feedback, supporting collaboration and building trust in environments where safety and reliability are essential.

3. From AI use cases to AI agents

The path to an agentic product development process (PDP) typically starts with a simple question: where can AI create the most value in your engineering workflows? From there, you can begin to connect individual use cases and evolve toward a coordinated, multiagent approach.

We already know that AI use cases deliver value. For example, at NTT DATA, we have identified 27 AI use cases across the full product development lifecycle, from early market analysis to maintenance and lifecycle management. These use cases show clear and proven benefits. AI can automate repetitive tasks, support analysis and help your teams work more efficiently across engineering activities.

[Explore the full set of AI use cases to see where you can create value across your engineering processes.](#)

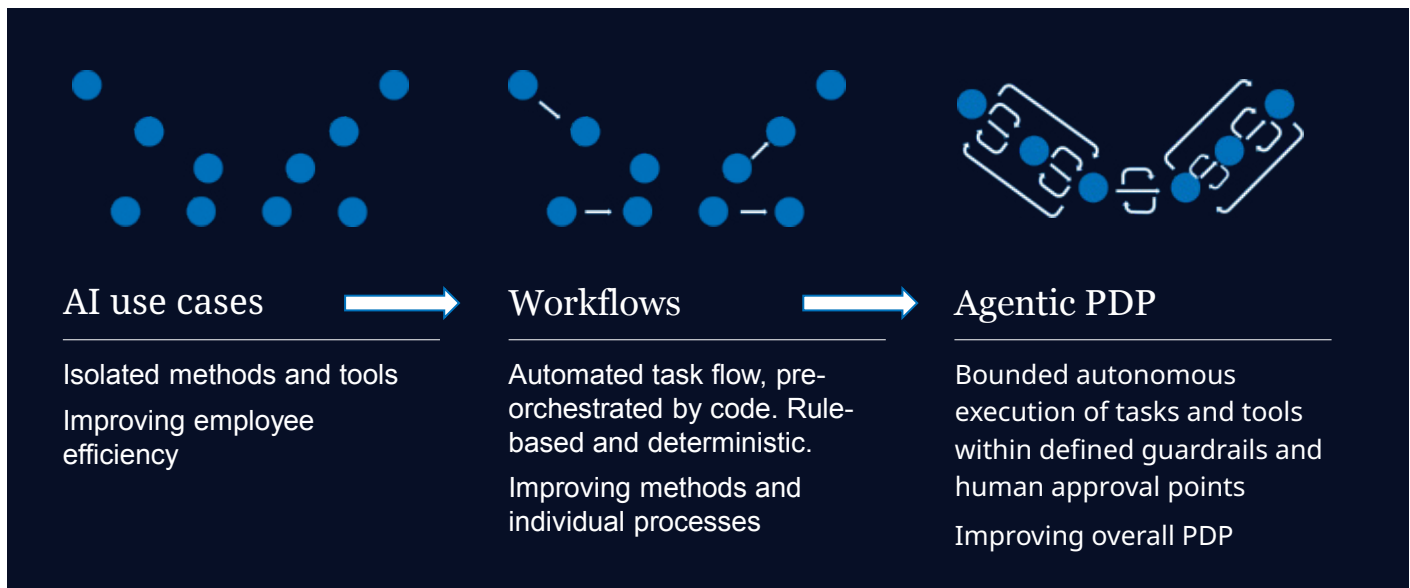
However, the real opportunity goes beyond individual use cases.

The true value comes from orchestrating these AI capabilities into connected, agentic processes. Instead of deploying isolated solutions, you can integrate AI across workflows, removing friction and enabling coordination between tasks, continuous data flow and faster, more informed decision-making.



3.1. Evolving from use cases to an agentic workflow

Moving to an agentic AI approach means shifting from isolated use cases to a coordinated, outcome-driven model. Instead of treating each AI capability separately, you can orchestrate them to support broader engineering goals. As this evolution progresses, AI moves from assisting individual tasks to enabling connected workflows, and ultimately to supporting an agentic product development process where AI agents dynamically execute tasks and use tools based on context, within defined guardrails, approval points and human accountability.



At this level, the impact goes beyond local efficiency gains. You can improve overall product development performance, accelerate innovation and respond more effectively to change.

At NTT DATA, we see this transition as guided by a set of key principles:

Specification-driven development

AI agents work best towards clearly defined goals. A clean set of high-level requirements is the basis for downstream activities such as architecture generation, design proposals or validation tasks. This ensures that AI remains aligned with engineering artifacts and reduces the need for manual coordination.

Model-based systems engineering integration

System models in SysML V2 provide a shared, machine-readable context and enable the “System as Code” approach. AI agents can use this to perform consistency checks, generate test cases and trace dependencies across the lifecycle. By connecting requirements, design, testing and operational data, you enable a continuous digital thread and support closed-loop development.

Process knowledge as reusable capabilities

Engineering processes are often documented but not directly executable. By structuring these processes into reusable Agent Skills, you allow AI agents to apply them dynamically. This makes workflows more flexible and adaptable, especially when real-world conditions do not follow predefined sequences.

Multiagent collaboration

Specialized agents can take on different roles, such as requirements analysis, design, testing or project coordination. An orchestration layer uses the reasoning power of modern LLMs and assigns tasks and consolidates outputs, ensuring alignment across the process. Standardized communication protocols make it easier to integrate both internal and third-party capabilities into a unified system.

Continuous learning and rapid iteration

By connecting agents in feedback loops, you can accelerate design and validation cycles. AI can explore multiple scenarios, evaluate outcomes and present the most relevant options. This supports earlier decision making and shifts development toward a more iterative and adaptive model.

3.2. Prioritization of agentic workflows and multiagent scenarios

Over the past years, NTT DATA has developed a clear perspective on where AI delivers the most value in engineering, guided by three criteria: business impact, feasibility and effort. These principles help identify use cases that are both practical and scalable.

Today, the focus is shifting from isolated AI use cases to orchestrated agentic workflows and multiagent systems. This transition introduces new dynamics, such as interdependencies and non-linear value creation, requiring a broader approach to prioritization that considers how use cases interact and scale across the product development process.

Top 8 agentic AI workflows

Number	Agentic workflow name
1	Requirements to test cases
2	Systems engineering zig-zag
3	Spec-driven development
4	Change analysis & implementation
5	Release management
6	Continuous compliance
7	Data-driven engineering
8	Variant optimization

3.2.1 Requirements → test cases (→ test execution → defects)

The process begins with extracting and structuring requirements, even when they come from unstructured or semi-structured sources.

From there, you:

#1

Transform requirements into clear, actionable test cases, including criteria and parameter sets

#2

Generate and execute test scripts across environments such as simulation platforms or hardware in-the-loop systems

#3

Capture defects and feed them back into the requirements and development process

This creates a continuous improvement loop, where each cycle strengthens quality, traceability and efficiency.



How AI agents support your teams

AI agents bring a new level of intelligence and coordination to this workflow.

Beyond simply scanning words, they interpret requirements using semantic reasoning, helping them understand intent and context. This allows them to:

- 1 Translate specifications into structured validation logic
- 2 Generate robust test designs automatically
- 3 Coordinate tools and testing environments across the lifecycle

They can also group defects and suggest potential root causes, so your teams can focus on resolving the issues that matter most.



What you need to make it work

1

Strong traceability between requirements, test cases and defects

2

Reliable access to testing environments, including simulation and hardware in-the-loop systems

3

A well-defined ontology to standardize how requirements and test concepts are structured

Challenges to plan for

Like any transformation, this approach comes with considerations.

If requirements are unclear or incomplete, test quality will suffer, so clarity at the start is key. In safety-critical contexts, full automation is not realistic, which means Human-in-the-loop validation remains essential. Approval gates must be maintained before safety-critical test execution, defect closure and release decisions, ensuring compliance with industry standards and organizational governance.

You may also face challenges integrating tools across your ecosystem. Many organizations still operate with fragmented toolchains, which can limit the effectiveness of end-to-end automation.

3.2.2 Systems engineering zig-zag (requirements → architecture)

This workflow aims to accelerate system design while ensuring consistency across all levels of architecture. It helps reduce misalignment between requirements and implementation, particularly in complex, multi-domain systems (mechanical, electrical, software).

The result is better system quality, fewer integration issues and improved collaboration across engineering disciplines.

This workflow follows an iterative systems engineering approach that connects top-down decomposition with bottom-up confirmation. Requirements are refined and allocated across functions, architectures, interfaces, subsystems and components, while architecture decisions are continuously checked against allocated requirements and traced back to stakeholder needs.

This bidirectional movement helps:

- **Maintain alignment across different levels of abstraction**
- **Improve consistency across domains**
- **Identify gaps early, before they impact integration**

How AI agents support your teams

AI agents help you manage complexity and move with greater confidence.

They can interpret and decompose requirements and generate system architectures such as SysML V2 models. They also support consistency checks across domains and levels, helping you keep everything aligned as the design evolves.

When you need to evaluate different options, agents can assist with trade-off analysis, giving you clearer insight into the impact of each design decision.



What you need to make it work

1

A mature model-based systems engineering environment

2

Shared ontologies that connect domains and teams

3

Clear versioning of architecture artifacts

These elements help ensure that your models remain consistent, traceable and scalable.

Challenges to plan for

Systems engineering brings inherent complexity, and automation has its limits.

The semantic depth of requirements and architectures can make full automation difficult. Dependencies across domains also increase coordination effort, especially in large-scale programs.

To succeed, you need a high level of data model maturity and a clear governance approach that keeps teams aligned as systems evolve.

3.2.3 Spec-driven development

This workflow focuses on compressing the embedded software development lifecycle by automating the transition from requirements to code and testing. It reduces manual handoffs and inconsistencies, enabling a shift from “AI-assisted engineering” toward more autonomous development workflows.

The process begins with capturing high-level intent or goals. From this, requirements, system architecture and code are generated.

Tests and documentation are automatically derived, and continuous validation ensures that outputs meet quality standards. The workflow is iterative, with ongoing reviews and refinements.



How AI agents support your teams

Agents perform multi-step reasoning across abstraction layers, generate code and validate it through feedback loops. They interact with development tools such as compilers, simulators and CI/CD pipelines, and perform self-checks to ensure quality.

What you need to make it work

Access to codebases, development standards (e.g. MISRA, AUTOSAR) and integration with CI/CD pipelines are essential.

Strong governance mechanisms to ensure quality and compliance.

Challenges to plan for

Trust in generated code is a major concern, especially in safety-critical systems. The orchestration of multiple steps and tools is complex, and strict validation processes are required to mitigate risks.

3.2.4 Change analysis & implementation

The process begins with a change request. From there, the system analyzes the impact across requirements, architecture, design, testing and project plans.

Based on this analysis, you can implement updates in a controlled way, ensuring that all affected elements remain aligned and traceable.

This approach helps you maintain consistency across the lifecycle while reducing the likelihood of rework.

Engineering changes are inevitable. What matters is how quickly and accurately you can assess their impact. This workflow helps you reduce cost and risk by improving the speed and precision of impact analysis.

When you understand how a change affects your system early, you can avoid unintended consequences, improve project predictability and keep delivery on track.

How AI agents support your teams

AI agents help you navigate complexity and make better decisions, faster.

They analyze dependency graphs and propagate impacts across related artifacts, giving you a clear view of how changes ripple through the system. They also support the planning and coordination of both technical and project activities.

To support decision-making, agents can simulate different scenarios and estimate cost, timeline and risk, helping you choose the most effective path forward.

What you need to make it work



A shared engineering ontology that connects requirements, architectures, interfaces, components, tests, defects, change requests and releases



Access to relevant artifacts and metadata across ALM, PLM, MBSE, test management and project systems, e.g. via MCP (Model Context Protocol)



The ability to combine existing trace links with AI-inferred dependencies where explicit links are missing



Baseline, version and ownership information so agents can assess which artifacts are affected by a change



Confidence scoring, evidence trails and validation rules to make agent recommendations explainable and reviewable

Challenges to plan for

This approach depends on the ability to reason across partial engineering context. Most organizations do not have complete traceability, so agents must combine available links with ontology-based dependency discovery, artifact semantics, architecture structures, interface definitions and change history. Impact assessments should clearly distinguish between confirmed, inferred and uncertain dependencies. Highly coupled systems can still make change propagation difficult to predict, and governance requirements may slow decision-making if not well aligned with delivery needs. Balancing transparency, engineering judgment and agility is key to making this workflow effective.

3.2.5 Release management

The workflow defines the release scope and continuously evaluates maturity across requirements, design, software, tests and engineering artifacts.

It identifies gaps, triggers necessary actions well before target milestones and validates the final release baseline against specifications and change requests.

It also generates documentation, version comparisons (diffs) and audit evidence.

The goal is to ensure reliable, predictable releases across complex systems while reducing integration risks and costly late-stage rework. It improves quality, compliance and time-to-market by enforcing structured validation of release readiness.

How AI agents support your teams

Agents assess completeness and consistency across artifacts, orchestrate workflows across tools and domains and manage dependencies. They can compare intended vs. actual system states and automatically generate reports and audit documentation.

What you need to make it work

- A shared release ontology connecting requirements, change requests, features, configurations, software versions, test results, defects, waivers, baselines and approval states
- Access to relevant artifacts and metadata across ALM, PLM, software repositories, test management, CI/CD and configuration systems
- The ability to combine existing trace links with inferred release dependencies where explicit links are missing
- Release readiness rules that define required evidence, open risks, blocking issues and acceptable deviations
- Confidence scoring and evidence trails so agent recommendations remain explainable and auditable

Challenges to plan for

Release decisions often depend on incomplete, distributed and changing evidence. Agents must distinguish between confirmed, inferred and uncertain release impacts, especially where explicit traceability is incomplete. Heterogeneous tools, inconsistent identifiers and conflicting baselines can make it difficult to compare intended and actual release scope. For critical releases, automation should be bounded by confidence levels, release policies and clear decision gates to avoid false confidence in readiness assessments.

3.2.6 Continuous compliance

The system continuously monitors regulatory updates and maps them to your engineering requirements.

As changes occur, it triggers the necessary updates across your lifecycle and generates the documentation and audit evidence you need.

This ensures that compliance is not a one-time activity, but an ongoing, integrated process.

Keeping up with evolving regulations can slow innovation and increase risk. This workflow helps you reduce compliance effort while accelerating certification processes, so your products stay aligned throughout their lifecycle.

When compliance is built into your engineering process, you can respond faster to regulatory changes and reduce the risk of delays or rework.

How AI agents support your teams

AI agents help you manage regulatory complexity with greater confidence.

They interpret regulatory texts and connect them to technical artifacts, ensuring that requirements remain aligned with current standards. They also generate supporting evidence and help enforce compliance policies across your processes.

This gives your teams clearer visibility and stronger control over compliance activities.

What you need to make it work

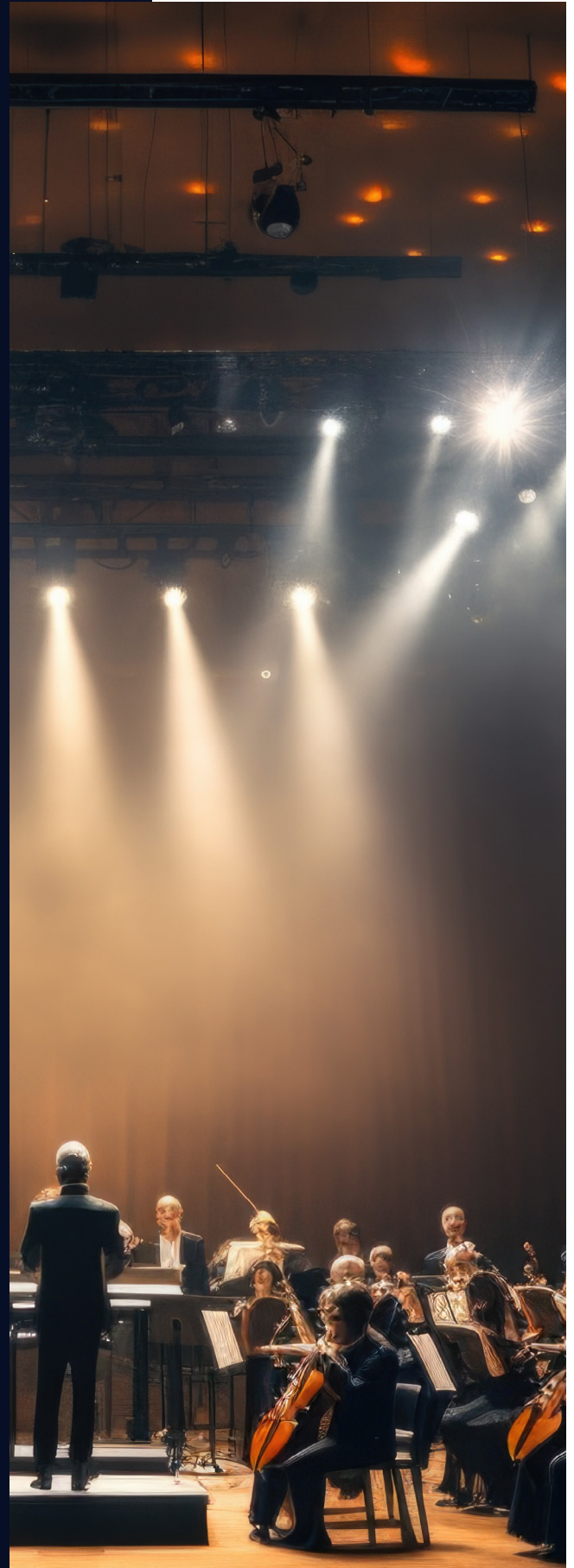
- A compliance ontology connecting regulations, obligations, standards, requirements, functions, tests, configurations, variants, markets and releases
- Access to regulatory content and internal compliance rules with provenance, applicability, version and effective-date metadata
- The ability to combine existing trace links with AI-inferred mappings between regulatory obligations and engineering artifacts
- An evidence model that defines which proof is needed, where it may exist and which baseline or release it supports
- Confidence scoring, source references and review rules to make compliance recommendations explainable and auditable

Challenges to plan for

Compliance decisions often depend on evolving regulations, product-specific applicability and distributed engineering evidence. Agents must distinguish between confirmed obligations, inferred impacts and uncertain interpretations, especially where explicit mappings to requirements, tests or configurations are incomplete. Regulatory sources, assumptions, confidence levels and evidence trails need to be transparent so high-impact findings can be reviewed without slowing down routine compliance work.

“

Agents must distinguish between confirmed obligations.”



3.2.7 Data-driven engineering (field → design loop)

This workflow closes the loop between products in the field and engineering teams in development. It uses real-world data from vehicles, service events, warranty claims and telemetry to detect quality issues, identify recurring patterns and feed insights back into product design.

The goal is to reduce warranty cost and recall risk while improving product quality based on how vehicles actually perform in operation. Instead of treating field issues as isolated incidents, this workflow helps you identify systemic problems earlier and translate them into targeted engineering actions.

By connecting field data with requirements, architecture, components, defects and change processes, you enable continuous product improvement across the lifecycle.

The process begins with collecting and harmonizing field, warranty, service and telemetry data. From there, the system detects patterns, anomalies and recurring failure modes across vehicle populations, variants, configurations and usage profiles.

These insights are then linked to potential defects, affected components, design decisions or validation gaps. Based on this analysis, engineering changes can be triggered, prioritized and tracked through the development process.

This creates a closed feedback loop where real-world product behavior informs future design decisions, validation strategies and quality improvements.

How AI agents support your teams

AI agents help you turn fragmented field information into actionable engineering insight.

They analyze large volumes of warranty claims, diagnostic trouble codes, service reports, sensor data and telemetry signals to detect patterns that may not be visible through manual analysis. They can identify anomalies, cluster similar issues and highlight correlations between failures, vehicle variants, environmental conditions, usage profiles and software or hardware configurations.

Agents can also generate root-cause hypotheses by linking field observations to engineering artifacts such as requirements, functions, interfaces, components, test cases, defects and previous change requests. This helps your teams move faster from symptom detection to engineering investigation.

Over time, agents support closed-loop learning by comparing predicted behavior with real-world outcomes. This allows validation strategies, design rules and future requirements to be improved based on operational evidence.

What you need to make it work



A field-to-engineering ontology that connects vehicles, variants, configurations, components, functions, requirements, tests, defects, service events, warranty claims and telemetry signals



Access to relevant field data, including vehicle data, diagnostic data, service records, warranty claims, customer issue reports and software or hardware configuration information



Variant and context awareness, so agents can distinguish between market, configuration, production batch, software version, usage profile and environmental effects



The ability to combine explicit links with AI-inferred relationships between field observations, defects, root-cause hypotheses and engineering artifacts



Integration with engineering systems such as ALM, PLM, MBSE, defect management, change management and quality systems



Confidence scoring and evidence trails so detected patterns, inferred causes and proposed engineering actions remain explainable and reviewable

Challenges to plan for

Field data is often incomplete, noisy, delayed or distributed across organizational silos. Warranty claims, service reports, telemetry signals and engineering defects may use different identifiers, levels of detail and terminology, making it difficult to connect symptoms in the field with design causes in development.

Root-cause attribution is another key challenge. A field issue may be caused by software behavior, hardware degradation, supplier variation, environmental conditions, user behavior, production deviations or interactions between multiple subsystems. Agents therefore need to distinguish between confirmed causes, likely hypotheses and uncertain correlations.

The workflow also depends on cross-functional collaboration. Quality, service, engineering, manufacturing, suppliers and product management often work in separate processes and systems. To make the field-to-design loop effective, agent recommendations should be supported by transparent evidence, confidence levels and clear ownership for follow-up actions. This helps teams act on real-world insights without creating false confidence in automatically generated conclusions.

3.2.8 Variant optimization

The system analyzes product variants using sales data, engineering complexity and cost insights.

Based on this analysis, it identifies optimization opportunities and feeds these insights back into product strategy and requirements.

This creates a continuous feedback loop that helps you refine your portfolio, reduce complexity and improve overall performance.

Managing product complexity can quickly increase costs and reduce margins. This workflow helps you simplify your product portfolio while improving profitability across the lifecycle.

By aligning product decisions with real data, you can reduce unnecessary variants, optimize configurations and focus on what delivers the most value to your business and your clients.

How AI agents support your teams

AI agents help you make better decisions across multiple dimensions.

They perform multi objective optimization and simulate trade-offs between cost, revenue and complexity. This gives you a clearer understanding of how different choices impact your business outcomes.

They also connect insights across engineering, sales and finance, helping your teams work from a shared view of the product landscape.

What you need to make it work

- Well-defined variant models
- Access to sales and cost data
- Integration across product lifecycle management, enterprise resource planning and sales systems

“ By aligning product decisions with real data, you can reduce unnecessary variants, optimize configurations and focus on what delivers the most value to your business and your clients.”

Challenges to plan for

Optimizing a product portfolio requires strong alignment across functions.

Engineering, sales and finance often have different priorities, which can make coordination more complex. Data fragmentation across systems can also limit visibility and slow decision-making.

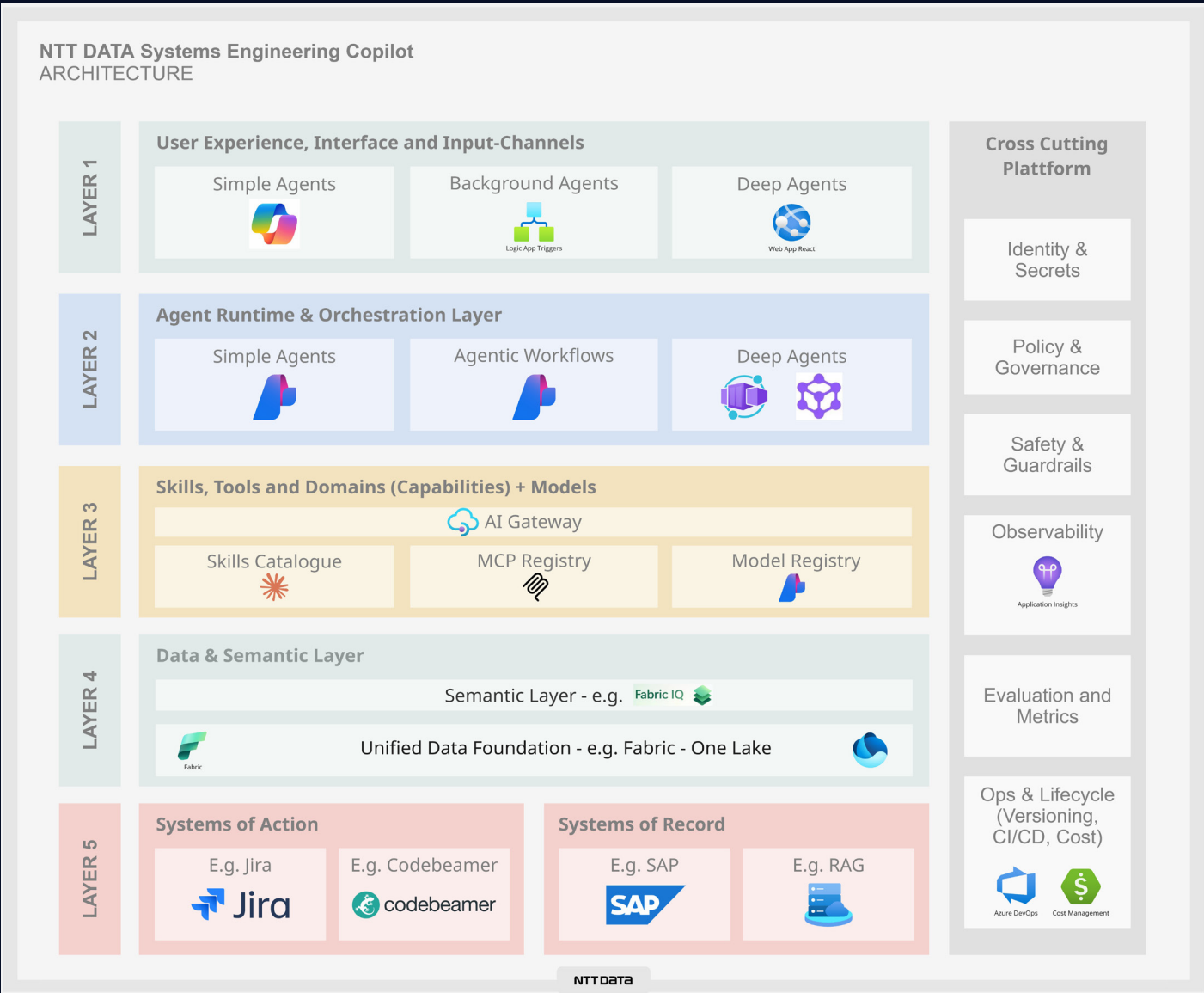
In addition, governance requirements may add constraints that need to be carefully managed to maintain both control and agility.

4. NTT DATA Systems Engineering Copilot

To address today's engineering challenges, you need more than a generic AI approach. You need a solution that addresses the specific challenges in complexPDP.

To operationalize an agentic approach to product development, NTT DATA has developed the Systems Engineering Copilot (SECP). This accelerator asset is essentially an agentic PDP environment built on Microsoft Azure, tailored specifically for the needs of systems engineering in industries like automotive. SECP is the flagship technical accelerator within NTT DATA's broader AI in Engineering offering.

With the Systems Engineering Copilot, NTT DATA provides that foundation. It supports the transition toward orchestrated agentic workflows across product development.



Source: NTT DATA

Deep integration with your engineering environment

The accelerator provides implemented and extensible integration patterns, including connections to systems such as PTC Codebeamer, product lifecycle management platforms, application lifecycle management tools like Jira and simulation environments.

At the same time, it understands engineering data models, including bills of materials, system architectures and test specifications. This allows you to embed AI into your workflows without rebuilding your landscape.

Built on scalable technologies such as Azure AI Foundry and extended with engineering-specific context and templates, the blueprint accelerates implementation and reduces effort from the start.

Orchestrated multiagent execution aligned with your goals

The NTT DATA Systems Engineering Copilot moves beyond the automation of individual tasks. It coordinates multiple agents that work together across your product development process.

You can define clear objectives, and the solution dynamically assigns tasks, monitors progress and adapts actions in real time. Agents collaborate across domains, enabling continuous workflows instead of disconnected activities.

This creates a structured, goal-driven execution model where AI supports both operational tasks and decision-making at scale.

Orchestration capabilities

Multiagent coordination	Reasoning	Agent registry	Context management
Human-in-the-loop	Tools & integration	Skills	Ontology
Model gateway	Governance	Security	Cost management

5 Implementing an agentic PDP

Shifting to an agentic AI-driven product development model is a transformative journey. To succeed, you need more than deploying technology. You need to identify high-value use cases, validate them quickly and scale them across your organization while managing change and governance.

At NTT DATA, we support you through this journey with a structured and practical approach. We work with you to define your strategy, demonstrate value early and build the foundations for long-term adoption.

Our roadmap includes a focused workshop to align priorities, a rapid proof of concept to show measurable results, an iterative path from pilot to production and the creation of an AI factory to scale capabilities. Along the way, we help you establish governance and change management practices so you can adopt AI confidently and sustainably.

5.1. Value discovery workshop – strategy and roadmap

A focused session to align your organization on where AI delivers the most value in engineering.

What you achieve

- Identify and prioritize high-impact agentic workflows
- Assess your current AI maturity and readiness
- Align business and engineering teams on a shared vision

How we approach it

- Prepare with targeted inputs and analysis of your engineering landscape
- Facilitate a structured workshop with key stakeholders
- Evaluate workflows based on impact, feasibility and effort
- Define clear value for each workflow with outcomes and success metrics

What you get

- A prioritized list of agentic workflows with business relevance
- A clear roadmap from proof of concept to next steps
- Alignment across teams to accelerate decision-making



5.2. Three-month value discovery PoC – quickstart implementation

A focused three-month initiative to validate AI workflows with real-world data and demonstrate measurable business value.

What you achieve

- Validate technical feasibility in your own environment
- Demonstrate tangible results with real engineering data
- Build a strong business case for scaling AI adoption

What you get

- Working prototypes integrated with your engineering workflows
- Quantified impact such as productivity gains and efficiency improvements
- Clear recommendations for scaling and industrialization

How we approach it

- Set up a secure AI environment
- Develop the agentic solution
- Iterate with stakeholders
- Measure business impact

Why it matters

- Reduces risk with a controlled and time-bound approach
- Provides evidence to support executive decisions
- Builds confidence by showing AI working with your data and processes

5.3. Establish an agentic AI factory: From PoC to production at scale

Once you validate value through the proof of concept, the next step is to scale and operationalize AI. This means moving from isolated use cases to a structured capability that continuously delivers, deploys and improves AI solutions across your engineering landscape.

With the Agentic AI Factory, NTT DATA helps you industrialize AI adoption and guide solutions through a clear path from prototype to production, ensuring long-term value and scalability.

What you achieve

- Scale proven use cases across multiple engineering domains, product lines, business units
- Transition from proof of concept to production-ready solutions
- Build a repeatable capability to develop, deploy and operate AI agents
- Accelerate innovation while maintaining control, governance and performance

How we approach it

- Building a cross-functional team
- Standardizing development and scaling
- Guiding the lifecycle from PoC to production
- Enabling foundations for AI operations and continuous improvement
- Embedding governance, compliance and security

What you get

- Production-ready AI solutions integrated into your workflows
- A scalable operating model for AI across engineering
- Faster delivery through reuse and standardization
- Continuous improvement based on real usage and evolving needs

Why it matters

- Moves you from isolated pilots to enterprise-wide impact
- Reduces time to value for new AI initiatives
- Ensures controlled, secure and compliant AI adoption
- Delivers sustained value through continuous improvement



5.4. AI governance and organizational change

Implementing agentic AI is not only about technology. It requires the right governance and a structured approach to organizational change to ensure long-term success. With NTT DATA, you combine technical innovation with the processes and cultural shifts needed to adopt AI with confidence.

What you achieve

- Ensure responsible, compliant and secure use of AI
- Build trust across engineering teams and stakeholders
- Drive adoption through clear processes and enablement

How we approach it

- Define governance structures, including Human-in-the-Loop governance models
- Ensure compliance and quality standards
- Validate and monitor AI performance
- Promote responsible and secure AI use
- Communicate clearly and build trust
- Train and empower your teams
- Adapt processes and responsibilities
- Provide ongoing support
- Measure adoption and impact

What you get

- A trusted and governed AI environment
- High adoption across engineering teams
- Sustainable and scalable transformation

Why it matters

- Reduces risk in regulated environments
- Builds confidence in AI-driven decisions
- Ensures long-term value beyond initial deployment

6. Authors



Jens Krüger

Executive Consultant

AI in Engineering



Frederik Bedrich

Head of Business Consulting

AI in Engineering



Julian Morelli Desanzo

Head of Technical Delivery

AI in Engineering

Visit nttdata.com to learn more.

NTT DATA is a \$30+ billion business and technology services leader in AI and digital infrastructure. We accelerate client success and positively impact society through responsible innovation. As a Global Top Employer, we have experts in more than 70 countries. NTT DATA is part of NTT Group.



