

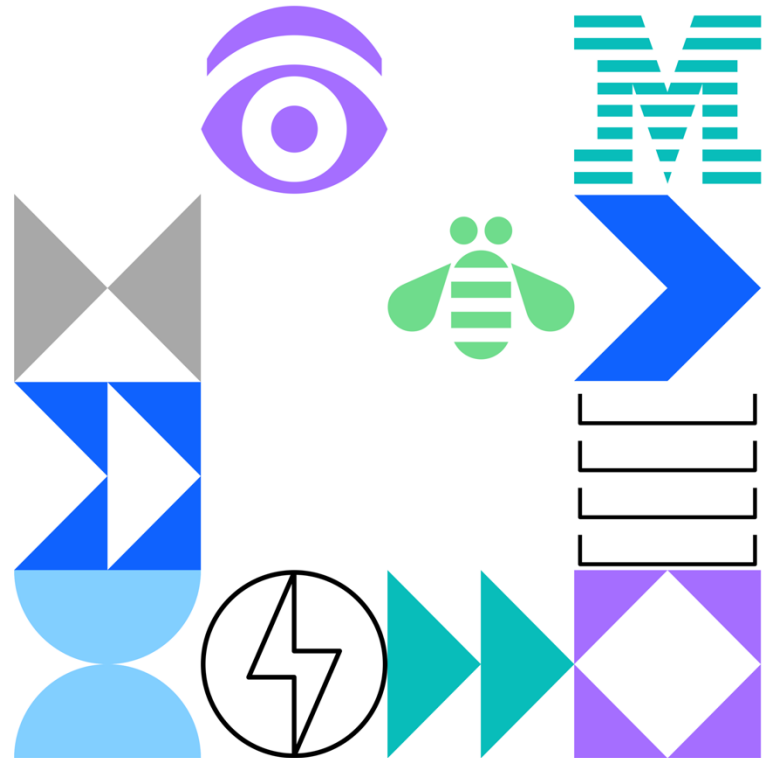
Transforming your PDLC and platforms with AI Agents

From coding assistants to
conversational agents in the PDLC and
software products

Pierre Feillet

Decision Intelligence Chief Architect – IBM

feillet@fr.ibm.com



IBM

Please note

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

NOTICE AND DISCLAIMER

- © 2026 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”
- Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

What we cover in this deck

01

Gen AI & Coding Agents

- Practitioner timeline: GenAI to agents
- Capability evidence: SWE-bench & METR
- Adoption signals: press, market & IBM Bob

02

The AI-Native Mindset

- What “AI-native” really means
- What's new for product teams
- Human vs AI contributions

03

Principles & Trust

- 5 Human-AI interaction principles
- Responsible & safe AI use
- Trust as a foundation

04

3 Stages of Adoption

- Stage 1 — Augment
- Stage 2 — Direct (test/spec)
- Stage 3 — Orchestrate

05

Building Conversational Agents

- How agentic systems work: architecture & tools
- IBM Bob, MCP & real-world examples
- Legal snapshot: AI Act (France/EU)

SECTION 01 of 05

Generative AI & Coding Agents

What industry practice show — across coding, business, and the product lifecycle

A practitioner's timeline: from GenAI to agents



Nov 2022 — ChatGPT & the LLM race

LLMs already existed (e.g., BERT); what changed was raw capability plus democratized access.



Reasoning models & chain-of-thought

Decomposing “reasoning” into multiple intermediate steps.



Tool use

Opening LLMs to the world — search, enterprise APIs, the Model Context Protocol (MCP).



Multi-agent & agentic systems

A ~40-year-old research thread (multi-agent systems, cybernetics) “rebooted” by LLMs; agent-to-agent (A2A) protocols.



Inflection point: coding agents

“Agents for coding like IBM Bob, Claude Code ... not in 10 years, but now” — one practitioner's framing.

From generation to agentic software engineering

Agentic systems plan, call tools, run tests, and iterate across many steps — instead of completing one prompt at a time.

◁> SWE-bench Verified — real GitHub issues resolved autonomously

~2%

Oct 2023
GPT-4 baseline

→

79.2%

Nov 2025
Claude Opus 4.5 + agent

Jimenez et al., 2023 (ICLR '24) · arXiv:2310.06770 — SWE-bench Verified leaderboard, Nov 2025

↗ METR — autonomous task time-horizon

Doubling ~every 7 months since 2019

Frontier models (e.g., o3) can now complete autonomous tasks that take skilled humans ~110 minutes, at 50% reliability.

Kwa et al., 2025 · arXiv:2503.14499 (METR)

METR frames this as an extrapolation, not a guarantee — but it is the clearest available signal of how fast agentic capability is moving.

Signs of adoption



External reporting increasingly matches the internal picture: adoption is accelerating and vendors are starting to disclose hard numbers — alongside the first controlled evidence that gains don't always hold up in practice.

CEO DISCLOSURES

~30%

of Microsoft's code (20-30%) — S. Nadella, Apr 2025

~25%

of Google's code — Google CEO, earnings call



25% of YC startups

have codebases ~95% AI-generated — TechCrunch, Mar 2025 batch



84% / 51%

use/plan to use AI tools, or use daily — Stack Overflow 2025 Survey



19% slower

COUNTERPOINT — METR RCT: experienced OSS devs were slower with AI, though they felt 20% faster (arXiv:2507.09089)



\$9.8–11B

Annualized enterprise AI coding agent market, Apr 2026 — Gartner

Most figures are company-disclosed or self-reported survey data, not independently audited — the METR figure is the exception, from a controlled study. See References for full citations.

The market: model providers, coding agents & DevOps tools



OpenAI

Frontier Model Lab

GPT models power coding agents, Codex CLI and ChatGPT's agent mode.



Anthropic

Frontier Model Lab

Claude models power Claude Code and many third-party coding agents.



IBM Bob

AI Assistant

IBM's agentic AI assistant for product & engineering work, via MCP.



Cursor

AI Code Editor

AI code editor with agent mode. Acquired by SpaceX for \$60B (Aug 2026).

DEVOPS TOOLS INTEGRATING AI ASSISTANTS



GitHub Copilot

In IDEs, PRs & Actions



GitLab Duo

Chat, code suggest & CI/CD



Atlassian Rovo

AI agents in Jira & Confluence



Jenkins

AI-assisted pipeline authoring



Docker

AI agent for Compose & debug



ServiceNow

Now Assist AI in ITSM flows

A Coding agent in the enterprise: IBM Bob



IBM Bob is an agentic AI development partner spanning the full SDLC. In the IDE it offers Agent, Ask, and Plan modes, spawns specialized subagents for parallel work, and extends via MCP and a standalone Bob Shell — with multi-model orchestration (Claude, Mistral, IBM Granite) and governance built in.

INTERNAL ROLLOUT

100 → **80,000+**

developers, June 2025 → April 2026

45% avg. self-reported productivity gain



70% reduction

IBM Instana team — time on selected tasks, ~10 hrs/week saved



69% time savings

IBM Maximo team — code generation & refactoring tasks



30 days → 3 days

Blue Pearl — Java upgrade; 160+ engineering hours saved, zero post-deploy defects



10× faster docs

APIS IT — legacy architecture documentation; .NET migration in hours vs. weeks

Figures are vendor-reported / self-surveyed, not independently audited. Sources: IBM Newsroom, "Introducing IBM Bob," Apr 28, 2026 · bob.ibm.com/docs/ide

SECTION 02 of 05

The AI-Native Mindset

Understanding what it means to work with AI — not just use it

What "AI-native" actually means

"Being AI-native isn't about using AI everywhere — it's about using AI with intention."

- AI accelerates speed, quality, and confidence in product discovery, build, and scaling – if taken with precaution
- People remain firmly in control of vision, strategy, and decision-making
- AI handles time-intensive tasks; humans focus on user empathy and ethical judgment
- Teams move faster and gain new capabilities without losing accountability

COMPLEMENTARY ROLES



Humans bring

- Meaning & judgment
- Responsibility
- Strategic vision
- User empathy



AI brings

- Speed & scale
- Exploration
- Pattern surfacing
- Execution tasks

What's new for product teams



Discovery

AI surfaces insights and accelerates research synthesis at scale

- Opportunity identification
- User research synthesis
- Competitive landscape
- Persona generation



Delivery

AI handles drafting, analysis, and generation so Product Managers focus on judgment

- Epic & story drafting
- Wireframe variations
- Technical spec generation
- Code generation
- Build
- Test scenario creation



Growth

AI monitors usage, surfaces anomalies, and generates weekly briefings automatically

- Dashboard auto-analysis
- Retention & adoption trends
- Automated weekly reports
- Funnel & conversion insight



Collaboration

AI integrates across tools so teams work from one connected workspace

- Jira, GitHub, Aha! connected
- Propel as unified layer
- No coding required
- MCP tool integrations

SECTION 03 of 05

Principles & Responsible AI

How humans and AI interact effectively — and safely

5 Principles for Human-AI Interaction

Successful human-AI collaboration enables both to learn and evolve over time.

1

Human-led intent & judgment

People lead direction and own decisions, refining goals through discovery while AI surfaces possibilities and tensions.

2

Shared sense-making of context

AI reveals patterns at scale while humans interpret meaning, co-creating understanding that evolves with context.

3

Outcomes as learning, not answers

Outputs invite reflection rather than conclusions. Surprise and disagreement drive learning through iteration.

4

Reflective creation & evaluation

AI analysis meets human judgment in iterative loops that preserve and blend complementary strengths.

5

Trust through clarity & resilience

Trust grows when AI outputs are inspectable, decisions are explainable, and data use is careful and robust.

Trust is foundational to the AI-native PDLC

Trust is **non-negotiable**. All AI-enabled work must follow organization's principles for safe, responsible and trusted AI use.



Keep outputs inspectable

All AI-generated outputs must be reviewable; keep decisions explainable to stakeholders



Handle data with care

Know your data sources, understand their limits, and protect sensitive information

⚠ PRACTICAL RULES FOR TEAMS

- ✓ Never ship AI output without human review
- ✓ Document AI-assisted decisions in artifacts
- ✓ Validate AI-generated data against source truth
- ✓ Apply your organization's AI ethics framework for user-facing features
- ✓ Maintain explainability at every decision gate

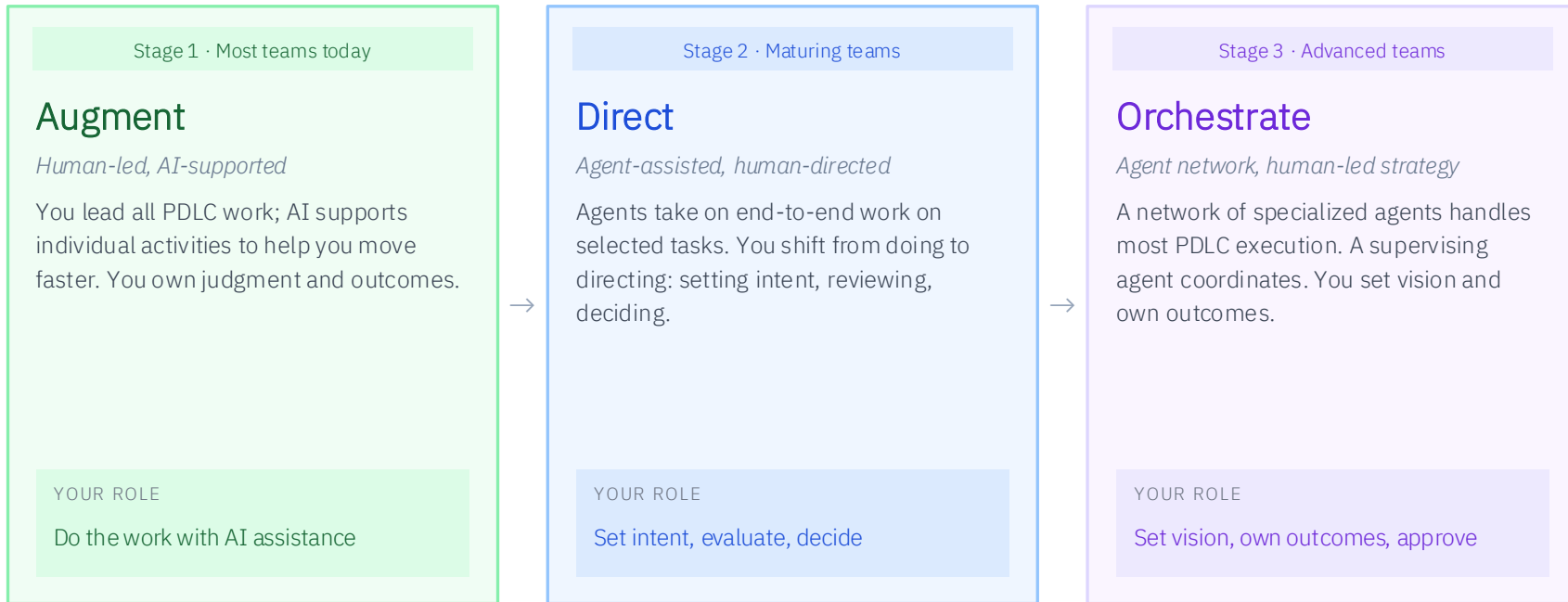
SECTION 04 of 05

3 Stages of AI-PDLC Adoption

From AI-assisted to AI-orchestrated — where is your team today?

Three stages of AI-PDLC adoption

Each stage shifts how much AI handles execution — and how much your role shifts from doing to directing.



Adoption and autonomy of agents are growing through the stages .

The AI-PDLC pipeline by maturity stage

Consolidated from the Stage 1–3 deep dives — requirements, then spec management, then delivery.

SPECIFICATIONS MANAGEMENT — 3 LEVELS, REFINED BEFORE CODE GEN

	Requirements	Specify	Plan	Tasks (Jira)	Code Gen (Git)	Code Review (PR)	Test	Validation	Deploy (CI/CD)
STAGE 1 Augment Human-led, AI-supported	AI 30% AI synthesizes research & transcripts; you validate themes	AI 15% You draft the spec from validated needs; AI flags gaps	AI 15% You design the architecture; AI offers alternatives	AI 35% Bob drafts epic + stories; you refine scope & confirm criteria	AI 10% You code from confirmed tasks; AI assists review, not gen	AI 30% AI adds code-review assist; you approve	AI 30% AI generates test scenarios; you write & run tests	AI 10% Manual security & compliance sign-off	AI 20% AI drafts launch prep & GTM content; you decide
STAGE 2 Guide Agent-assisted, human-directed	AI 60% Agent identifies opportunities & assesses solutions; you validate	AI 65% You describe the outcome; agent drafts spec around user journeys	AI 60% You add constraints — stack, architecture; agent drafts blueprint	AI 75% Agent decomposes the plan into reviewable, testable chunks	AI 80% Agent executes tasks in sequence — focused diffs, not one big change	AI 50% You review outputs critically; judgment separates good from automated	AI 70% Agent builds & tests the solution autonomously	AI 35% You validate UX & accessibility before implementation proceeds	AI 55% Agent launches at scale with your approval
STAGE 3 Orchestrate Agent network, human-led strategy	AI 70% Discovery agent researches within the intent you set	AI 65% Agents derive the spec from your stated vision	AI 85% Supervising agent coordinates the blueprint across specialists	AI 90% Supervising agent manages backlog & handoffs end-to-end	AI 90% Build agent implements from the coordinated plan	AI 70% Test agent verifies; you apply judgment at key control points	AI 90% Test agent runs & maintains suites autonomously	AI 40% You validate usability & accessibility — still yours to own	AI 75% Launch + Analytics agents ship & measure; you approve what ships

What's needed is captured once, then Specify → Plan → Tasks turn it into a build-ready spec.

AI % is indicative — the share of that step's output produced by the agent — not a measured metric.

Stage 1: Augment — you lead, AI helps

AI SUPPORTS ACROSS ALL PDLC PHASES

🔍 Discovery

Research synthesis, persona drafts, competitive analysis, opportunity framing

✍️ Definition

Epic & story drafting, acceptance criteria, prioritization support

🏗️ Build & Test

Wireframe variations, test scenario generation, code review assist

🚀 Launch & Scale

Launch prep, dashboard analysis, GTM content, usage insight

YOUR ROLE STAYS CENTRAL

You own judgment calls and outcomes. AI handles time-intensive tasks. Your expertise evaluates what best serves users.

STAGE 1 IN PRACTICE

Discovery

AI synthesises 20 interview transcripts → you evaluate themes and validate against user needs

✓ Human: selects which themes matter

Delivery

Bob drafts Epic + 8 user stories → you review, refine scope, confirm acceptance criteria

✓ Human: approves what ships

Growth

Amplitude AI generates weekly usage report → you interpret trends, decide action

✓ Human: owns the growth decision

Stage 2: Direct — agents work, you guide

Variation A: Test-driven PDLC

Rapid experimentation

Development happens through rapid experimentation and feedback loops. Agents handle end-to-end work:

- Identify opportunities and assess solutions
- Build and test the solution autonomously
- Launch at scale with human approval

Variation B: Spec-driven PDLC

Spec-first build

Difference shows up in the build stage. Agents generate highly detailed technical specs before any implementation:

- Agent generates comprehensive technical spec
- You review, validate UX, accessibility standards
- Implementation begins only after your approval

YOUR ROLE IN STAGE 2



Set intent clearly

Define what success looks like; agents need clear goals to produce good outputs



Review outputs critically

Your judgment separates a good outcome from an automated one



Make key decisions

Refine direction at each decision point; you own the final call



Validate UX & accessibility

Spec-driven: check agent specs against usability and accessibility requirements

Stage 3: Orchestrate — you set intent, agents execute

A network of specialized agents handles most PDLC execution. A supervising agent coordinates tasks, manages handoffs, and ensures outputs come together into a complete flow.

YOUR ROLE — THE MOST STRATEGICALLY IMPORTANT

- | Set the intent that drives everything
- | Own the outcomes that result
- | Apply judgment at key control points
- | Define the experience vision
- | Validate usability & accessibility
- | Approve what ships to users



You (Product Manager) — Strategic lead

Set intent → validate outputs → approve decisions



Supervising Agent

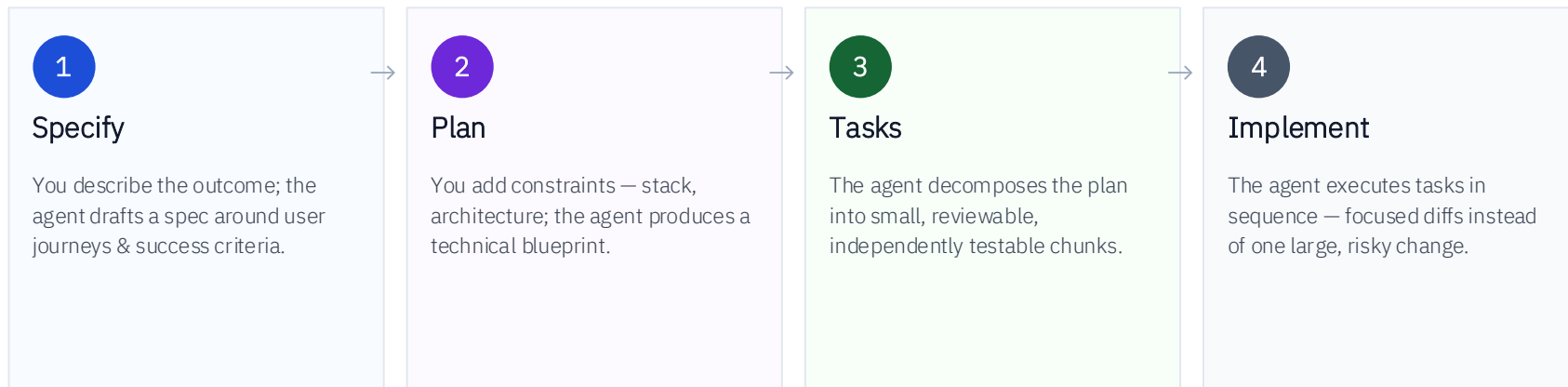
Coordinates tasks, manages handoffs across specialist agents



Specialist Agents

Discovery agent · Build agent · Test agent · Launch agent · Analytics agent

Spec-driven development: the 4-phase workflow



WHY IT BEATS “VIBE CODING”

Fewer unstated requirements

Clear specs stop agents guessing at requirements nobody wrote down.

No silent quality trade-off

Microsoft research: vibe coding's speed gain came with more complexity & static-analysis warnings.

Verifiable at each step

Tasks are discrete & testable — mistakes are caught before they compound.

Tooling: GitHub Spec Kit (open source), Claude Code (CLAUDE.md), Cursor rules, and similar spec-first workflows.

SECTION 05 of 05

Building Conversational Agents

From agent architecture to IBM Bob, MCP servers — and legal guardrails

Agent Anatomy

Definition

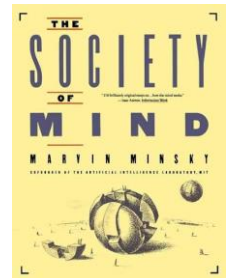
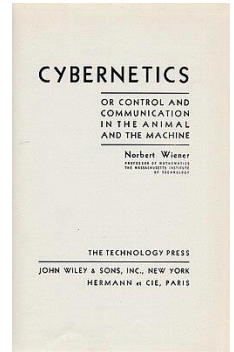
- Agentic AI = AI systems that **act autonomously to achieve goals** with limited human supervision
- Built from **AI agents** that mimic human decision-making
- Often organized in **multi-agent systems** coordinated via orchestration

Key Characteristics

- **Autonomous** → operates without constant human input
- **Goal-driven** → plans and executes multi-step tasks
- **Adaptive** → learns from feedback and improves over time
- **Proactive** → combines LLM flexibility with structured decision-making

Core Capabilities

- Interacts with tools (APIs, databases, web)
- Specializes into roles (simple → complex agents)
- Enables natural language interaction (no complex UI needed)



The Agent Loop: How Agents Operate

How Agentic AI Works (Lifecycle)

- Perception → collect data (APIs, sensors, users)
- Reasoning → interpret context & extract insights
- Goal Setting → define objectives & strategy
- Decision-Making → choose optimal action
- Execution → act via systems or responses
- Learning → improve via feedback
- Orchestration → coordinate multiple agents

Example Use Cases

- IBM Bob — codes, tests, refactors
- IBM ODM — checks loan eligibility via MCP
- Decision Assistant — drafts rules from text
- Amplitude AI — weekly usage insights
- MCP chat agent for support & research

Key Challenges

- Misaligned goals → unintended behaviors
- Reward hacking (e.g., optimizing wrong metrics)
- Multi-agent complexity → bottlenecks, conflicts

Extending a generic assistant into an agent



MCP

Model Context Protocol

Open standard connecting an assistant to external tools, APIs & data.



Agent Skills

Packaged know-how

Instructions, scripts & resources an assistant loads on demand for a task.



A2A

Agent2Agent

Lets independent agents discover each other and collaborate as peers.



Subagents

Orchestration

A supervising agent spawns specialists for focused, parallel work.

OTHER EXTENSION MECHANISMS



Modes

Constrained styles (Agent / Ask / Plan) scoping what it may change.



Memory & context

Persistent state across sessions: prior decisions & project history.



Tool / function calling

The foundational primitive underneath MCP: calling a named function.

The building blocks: Bob, Propel, GitHub and Amplitude



IBM Bob

AI Assistant

IBM's agentic AI assistant for product & engineering work, queryable via MCP.

- Draft epics, stories, PR-FAQs
- Synthesise research & interviews
- Run skills: Jira, Aha!, web research



Propel

AI-Native Workspace

Purpose-built workspace on Bob for product teams. No coding required.

- Jira, Aha!, Amplitude connected
- Skills for Discovery, Delivery, Growth
- Product knowledge persists across sessions



GitHub

Source Control + CI/CD

Where code, issues, and pull requests live — connective tissue across the toolchain.

- PRs reviewed, tested & merged
- CI/CD pipelines triggered
- Issues synced with Propel & Bob



Amplitude AI

Product Analytics + AI

Analytics platform with built-in AI for natural-language questions and agents.

- Home AI chat for instant answers
- Custom agents for weekly reports
- AI side-panel on any dashboard

IBM Bob

Agentic Coding Assistant

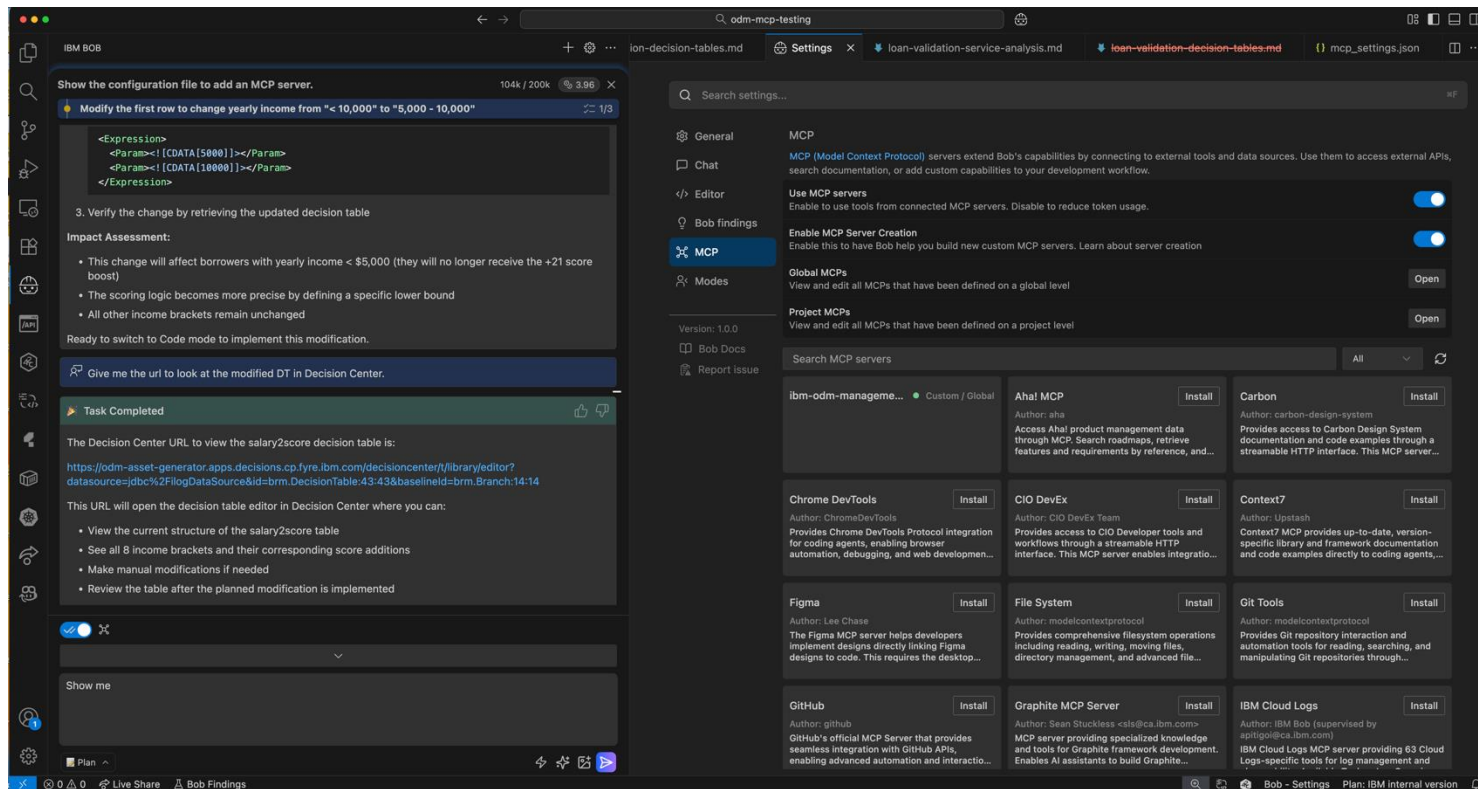
In VSCode

CLI

Open to MCP servers

Configuration through Mods

Towards an Agentic Cockpit for DevOps



Example of Embedded assistant in IBM Decision Intelligence

Loan Approval Decision Service

Step 1: Setting the scope | Step 2: Building the structure | Step 3: Defining the rules | Step 4: Finalizing the service

Policy | Data model | Decision model

Initial Policy Prompt
Approve a loan

Loan Approval Policy (US)

Purpose:
To establish clear criteria for the approval or rejection of loan applications for customers in the United States to ensure consistency and mitigate risk.

Scope:
This policy applies to all individual loan applications submitted to Acme Bank within the United States.

Eligibility Criteria
To be considered for a loan, applicants must meet the following requirements^[1]:

- Age: Applicants must be at least 18 years old^[2].
- Residency: Must be a resident or citizen of the United States^[3].
- Credit Score: Minimum credit score of 600^[4].
- Applicants with credit scores below this threshold may be rejected or offered alternative loan terms (e.g., higher interest rates or co-signer requirements).
- Income:

References

ELIGIBILITY CRITERIA
"To be considered for a loan, applicants must meet the following requirements."

Age
"Applicants must be at least 18 years old"

RESIDENCY
"Must be a resident or citizen of the United States"

IBM Decision Intelligence

Decision Assistant

Co-Signers:
If the applicant does not meet the credit score or income criteria, they may apply with a co-signer who meets the necessary qualifications.

Manual Review:
In cases where the applicant's situation is not clearly addressed by the criteria, a manual review by a senior loan officer may be conducted.

Review and Updates:
This policy is subject to periodic reviews and updates as needed to align with regulatory requirements and market conditions.

with 8:46 AM
Does the data model look correct to you? Check that it reflects the decision you want to automate.

Yes Edit Regenerate

with 8:47 AM
Step 2 of 4 completed!

Loan Approval Decision Service
A decision service to evaluate loan applications based on eligibility criteria and risk assessment as per Acme Bank's Loan Approval Policy. Assumptions made for missing details include the use of credit score and debt-to-income ratio as primary risk assessment factors.

Viewing

Is this how you expect the decision to behave? Review the generated model to ensure it aligns with your expectations.

Yes Edit Regenerate

Loan Approval Decision Service

Step 1: Setting the scope | Step 2: Building the structure | Step 3: Defining the rules | Step 4: Finalizing the service

Policy | **Data model** | Decision model

Data Model

LoanApplication → EligibilityCheck → CreditworthinessAssessment → LoanTermsDetermination → LoanDecision

LoanApplication → EligibilityCheck → LoanDecision

LoanApplication → CreditworthinessAssessment → LoanTermsDetermination → LoanDecision

LoanApplication → CreditworthinessAssessment → LoanDecision

LoanApplication → LoanTermsDetermination → LoanDecision

LoanApplication → LoanDecision

LoanApplication
Input parameter containing details of the loan application^[5]

EligibilityCheck
Evaluates if the applicant meets the basic eligibility criteria^[6]

Name: CheckAgeEligibility
Rule: If the age of LoanApplication is at least 18 then set decision to a new EligibilityCheck where ageEligible is true;

Name: DefaultAgeEligibility
Rule: set decision to a new EligibilityCheck where ageEligible is false;

Name: CheckResidencyEligibility
Rule: If the residencyStatus of LoanApplication is "United States" or the residencyStatus of LoanApplication is "Resident" then set decision to a new EligibilityCheck where residencyEligible is true;

Name: DefaultResidencyEligibility
Rule: set decision to a new EligibilityCheck where residencyEligible is false;

Name: CheckIncomeEligibility
Rule: If the annualIncome of LoanApplication is at least 30000 then set decision to a new EligibilityCheck where incomeEligible is true;

Name: DefaultIncomeEligibility
Rule: set decision to a new EligibilityCheck where incomeEligible is false;

Name: CheckEmploymentEligibility
Rule: If the employmentStatus of LoanApplication is "Employed" or the employmentStatus of LoanApplication is "Self-Employed" then set decision to a new EligibilityCheck where employmentEligible is true;

MCP Servers

Model Context Protocol

A de facto standard to plug any set of APIs in an LLM and agentic platform (wxO, Claude, etc)

<https://www.ibm.com/think/topics/model-context-protocol>

An MCP server publishes a set of tools, resources and prompts

- Can be local or remote

An example: The IBM ODM Decision MCP Servers bridges IBM ODM with modern AI assistants and orchestration platforms.

- a ready to use Open Source MCP server to execute symbolic reasoning
- a ready to use Open Source MCP server for governing Decision Service projects

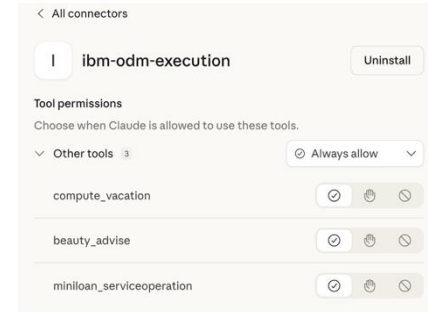
It enables you to:

- Expose decisions as tools for AI assistants
- Automate decisions dynamically in workflows
- Centralize and expose business logic to end users and bots

Features

- Authentication: Zen API Key, Basic Auth, and OpenID Connect
- Multi-Platform: Works with Watson Orchestrate, Claude Desktop, IBM Bob and Cursor AI

Video: [Rule Execution Demo in Claude](#)



Legal risks of using AI code generators



IP & Copyright

Generated code can echo copyrighted training data — whose use for training is still being litigated.



Open-Source License Risk

Output can reproduce GPL/AGPL fragments, triggering copyleft disclosure duties.



Product Liability (EU)

Since Aug 2026, EU law treats you as the code's 'producer' — strict liability, reversed burden of proof.



Confidentiality & GDPR

Cloud AI tools risk exposing trade secrets; CNIL flags agent memory & traceability gaps.

EU AI ACT & FRANCE — STILL THE BACKDROP

Transparency duty

Art. 50: users must be told they're interacting with an AI.

Phase-in timeline

High-risk obligations: Dec 2027 (autonomous) / Aug 2028 (regulated products).

CNIL agentic AI note (Jul 2026)

Traceability, memory limits & human-in-the-loop for autonomous actions.

What to remember

The most important ideas from this training

- 1 Intent beats ubiquity**
AI-native ≠ AI everywhere. Use AI where it creates real value; preserve human judgment where it matters most.

- 2 Humans own outcomes**
At every stage — augment, direct, or orchestrate — humans set vision, own decisions, and are accountable for results.

- 3 Trust must be designed in**
Keep outputs inspectable, decisions explainable, data use careful. Trust grows through clarity, not assumption.

- 4 Know your adoption stage**
Most teams are Stage 1. Understand where you are; build toward Stage 2 with intent, not pressure.

- 5 AI doesn't change what — it changes how**
The PDLC still has Discovery, Delivery, and Growth phases. AI accelerates execution within them.

- 6 Compliance is part of the build**
AI Act obligations phase in through 2027-2028. Check your agent's risk tier, keep a human in the loop, and involve Legal early.

Sources cited in this deck, plus further reading

ACADEMIC & RESEARCH

Chen et al., 2021 — Codex & HumanEval

[arXiv:2107.03374](https://arxiv.org/abs/2107.03374)

Peng et al., 2023 — GitHub Copilot RCT

[arXiv:2302.06590](https://arxiv.org/abs/2302.06590)

Jimenez et al., 2023 (ICLR '24) — SWE-bench

[arXiv:2310.06770](https://arxiv.org/abs/2310.06770) · SWE-bench Verified leaderboard, Nov 2025

Kwa et al., 2025 — METR long-task time horizons

[arXiv:2503.14499](https://arxiv.org/abs/2503.14499)

METR, 2025 — Experienced OSS developer RCT

[arXiv:2507.09089](https://arxiv.org/abs/2507.09089)

Adoption & Impact of Command-Line AI Coding Agents (Microsoft study)

[arXiv:2607.01418](https://arxiv.org/abs/2607.01418)

IBM BOB & IBM SOURCES

IBM Bob

bob.ibm.com

IBM Bob — IDE documentation

bob.ibm.com/docs/ide

IBM Newsroom — “Introducing IBM Bob,” Apr 2026

newsroom.ibm.com

IBM Decision Intelligence · What is Agentic AI

ibm.com/products/decision-intelligence · ibm.com/think/topics/agentic-ai

REGULATORY, MARKET & SECURITY

Règlement (UE) 2024/1689 — AI Act

eur-lex.europa.eu/eli/reg/2024/1689

EU Digital Omnibus — high-risk AI deadlines postponed

digital-strategy.ec.europa.eu

Verizon 2025 / 2026 Data Breach Investigations Report

verizon.com/business/resources/reports/dbir

CISA BOD 19-02 · 22-01 · 26-04

cisa.gov/news-events/directives

Cloud Security Alliance — “The Collapsing Exploit Window,” 2026

labs.cloudsecurityalliance.org

Coding-agent landscape roundups, 2026

Value Add VC · MarkTechPost · Faros AI · Levelop

INDUSTRY, PRACTITIONER & TRAINING

Model Context Protocol

ibm.com/think/topics/model-context-protocol

BeeAI

beeai.dev

Agentic AI courses (DeepLearning.AI)

learn.deeplearning.ai/courses/agentic-ai

Wrap-Up

1

Agentic AI is real and is disrupting software industry and business

2

- Agentic coding stacks like IBM Bob and Claude + MCP/Skills are game changers in DevOps
- Re-think your Software Factory and PDLC

3

Expect new players, in-house and new competitors to emerge in every field at a high speed and with disruptive business model

4

- Be curious, pragmatic and adapt fast
- Classic way to develop features becomes obsolete
- Moving from +AI to AI+ while keeping Trust at the center

Thank you!

Co-Innovate with

