

The article "**The End of Coding as We Know It**" (published in *The New York Times Magazine* on March 12, 2026) explores the profound transformation of the software engineering profession triggered by the arrival of "agentic" AI tools like **Claude Code** and **ChatGPT**.

Core Argument

The piece argues that 2026 marks a tipping point where AI has transitioned from a "copilot" that suggests snippets of code to an "agent" capable of independently managing entire software projects. The central thesis is that the traditional identity of a "programmer"—someone who writes syntax—is being replaced by the "builder" or "product manager," where the primary skill is system design and intent communication rather than manual coding.

Key Themes and Insights

- **The Rise of Agentic AI:** Unlike earlier LLMs that required human intervention for every step, the article highlights how tools like Anthropic's **Claude Code** can now autonomously write tests, run them, identify bugs, and iterate until the code works. This "closed loop" removes the human from the tedious execution phase of development.
- **The "Junior Developer" Crisis:** A significant portion of the article focuses on the labor market. With AI able to handle entry-level tasks with near-perfect efficiency, the traditional "apprenticeship" model of software engineering is breaking. Companies are hiring fewer junior developers, raising concerns about where the next generation of senior architects will come from if they never learn the "hard way."
- **The Democratization of Building:** Conversely, the article captures the optimism of non-technical creators. By lowering the barrier to entry, AI is allowing entrepreneurs and "solopreneurs" to build complex applications that previously required venture-backed engineering teams.
- **Economic and Existential Shifts:** Interviewing industry figures like Boris Cherny (Chief Architect of Claude Code), the article touches on the "painful" transition ahead. It suggests that while total productivity may skyrocket, the specific job title of "Software Engineer" may become a relic of the past, replaced by roles that prioritize high-level problem-solving and ethical oversight.

Conclusion

The article concludes that while coding isn't "dying," it is becoming invisible. Much like how compilers replaced assembly language, AI is becoming the new layer of abstraction, leaving humans to focus on *what* to build rather than *how* to write the syntax to make it happen.

OUTSIDE article: how to prepare?

As we move through 2026, the role of an IT professional has shifted from **writing** code to **orchestrating** systems. Because AI agents (like Claude Code and GitHub Copilot) can now handle the syntax and boilerplate, the value of a human engineer has moved "up the stack."

Here is a roadmap for IT students and engineers to remain indispensable in an AI-dominant era:

1. Master "Agentic" Orchestration and Tooling

In 2026, you aren't just using a chatbot; you are managing a fleet of AI agents. You must understand how these agents interact with real-world environments.

- **Agentic Frameworks:** Study how to build and maintain autonomous loops (e.g., LangChain, AutoGPT, or specialized internal agent frameworks).
- **AI Infrastructure:** Learn how to set up the environments where AI agents live—Docker containers, sandboxed execution environments, and automated CI/CD pipelines that AI can trigger.
- **Prompt Engineering vs. Intent Engineering:** Move beyond simple prompts to "System Instructions" and "Few-shot prompting," where you provide the AI with complex architectural constraints rather than just task descriptions.

2. Deepen Systems Architecture and Design

AI is excellent at writing functions but still struggles with the "big picture" of how a massive system hangs together.

- **System Design:** Study how to scale distributed systems, microservices, and cloud-native architectures. You need to be the "Architect" who tells the AI how to connect the blocks.
- **Legacy Integration:** A massive part of the job will be using AI to refactor and maintain old codebases. You need to understand how modern AI layers sit on top of legacy SQL databases or COBOL systems.

3. AI Safety, Security, and Observability

As AI writes more of the code, the risk of "hallucinated" security vulnerabilities increases. Humans must become the ultimate "Auditors."

- **AI Security (AISec):** Learn how to protect against prompt injection, data poisoning, and insecure code generation.
- **Observability:** Study tools like Prometheus, Grafana, or AI-specific monitoring (like Weights & Biases) to track what the AI is actually doing in production.
- **Formal Verification:** Learning how to mathematically prove that code is correct will become more valuable than the manual act of writing the code itself.

4. Data Engineering and RAG (Retrieval-Augmented Generation)

AI is only as good as the data it can access. Modern engineers must know how to feed the AI the right context.

- **Vector Databases:** Master databases like Pinecone, Milvus, or Weaviate.
- **Data Pipelines:** Learn how to clean and structure proprietary company data so that an AI can use it to provide accurate answers (RAG).
- **ETL (Extract, Transform, Load):** Understanding how to move data from point A to point B remains a core skill for any AI-integrated system.

5. The "Human" Skills (Product & Ethics)

In 2026, the "soft" skills have become "hard" skills. If the AI can build anything, the question is: *What is actually worth building?*

- **Product Management:** Learn how to define user needs and translate them into technical requirements. Engineers are becoming "Product Builders."
- **AI Ethics and Compliance:** Study the legal frameworks (like the EU AI Act) and ethical considerations regarding bias and displacement.
- **Communication:** Your ability to explain technical trade-offs to non-technical stakeholders is now your most valuable asset.

Summary Checklist for 2026:

- **From:** Coding Syntax \rightarrow **To:** Reviewing and Auditing Code.
- **From:** Manual Debugging \rightarrow **To:** Designing Automated Test Suites.
- **From:** Building Features \rightarrow **To:** Designing System Workflows.
- **From:** Local Development \rightarrow **To:** Cloud-Agent Orchestration.

AND

what about requirements of it system and very big it systems or critical systems: humans are needed?

The short answer is **yes**, but the human role has shifted from "construction worker" to "building inspector and structural engineer."

In 2026, as systems become more "agentic" (AI-driven), the stakes for failure in critical infrastructure—banking, healthcare, power grids, or aerospace—have actually made human oversight **more** vital, not less.

Here is why humans remain the "fail-safe" for large-scale and critical IT systems:

1. The "Ambiguity" Gap in Requirements

AI is a literalist. It excels at executing a defined task but struggles with the conflicting, messy, and political nature of human requirements.

- **Stakeholder Negotiation:** In a "very big" system, different departments often have contradictory needs. A human engineer must navigate the office politics, legal constraints, and budget trade-offs to decide what the system *should* do.
- **Edge Case Discovery:** AI can only predict based on patterns it has seen. Humans are needed to imagine "Black Swan" events—unprecedented scenarios that haven't happened yet but would be catastrophic if they did.

2. Liability and "The Throat to Choke"

In critical systems, there is a legal and ethical requirement for **accountability**.

- **Moral Responsibility:** If an AI-driven medical system misdiagnoses a patient, a "hallucination" isn't a legal defense. A human professional must sign off on the architecture to certify it meets safety standards.
- **Regulatory Compliance:** Large systems must adhere to strict frameworks (like the EU AI Act or HIPAA). Humans are needed to interpret how these evolving laws apply to a specific technical implementation.

3. Verification of "Black Box" Logic

Deep learning models can be "black boxes"—they provide an output, but they don't always explain the *why* behind it.

- **Formal Verification:** For critical systems (like flight control software), we use mathematical proofs to ensure the code cannot enter a "fail state." Humans must design these proofs and interpret the results.

- **The "Human-in-the-Loop" (HITL) Requirement:** Many critical systems are now designed so that the AI proposes a solution, but a human must click "Execute." This prevents a runaway "automated catastrophe."

4. Resilience and "Graceful Degradation"

When an AI agent fails, it often fails "brittley"—it might continue to produce confident but wrong results.

- **Disaster Recovery:** If the AI infrastructure itself goes offline, humans need to know how to operate the system manually or trigger "safe mode" protocols.
- **Systemic Over-reliance:** A major risk in 2026 is "Automation Bias," where humans stop paying attention because the AI is usually right. Engineers are now being trained specifically to *distrust* the AI to maintain high-alertness levels for critical systems.

Comparison: Human vs. AI in Large Systems

| Task | AI Agent Role (The "Doer") | Human Engineer Role (The "Owner") |
|--------------|---|--|
| Requirements | Generates user stories based on input. | Defines the "Definition of Done" and ROI. |
| Security | Scans for known vulnerabilities (CVEs). | Models "Social Engineering" and zero-day threats. |
| Scaling | Automatically provisions cloud resources. | Monitors cost-efficiency and long-term sustainability. |
| Failure | Restarts services or reroutes traffic. | Conducts Post-Mortems and addresses root causes. |

AND

Here is how IT programs have changed and the new "success profile" for students in this era.

1. How IT Programmes Have Changed

University and technical curricula have been restructured to focus on **Systemic Understanding** rather than **Syntactic Mastery**.

- **From Syntax to Orchestration:** Standard "Intro to Programming" courses (Java, C++, etc.) that spent months on semicolon placement have been replaced by "Computational Logic and Agent Orchestration." Students now learn to use **Agentic Frameworks** (like LangChain or Anthropic's MCP) to coordinate multiple AI agents to build apps.
 - **The "Vibe Coding" Reality:** Lab assignments now assume the use of AI. Instead of "Write a function that sorts an array," the assignment is "Direct an AI agent to build a secure, scalable healthcare dashboard and then **mathematically verify** its security."
 - **Continuous Feedback Loops:** High-stakes final exams are being replaced by "Formative Assessments." AI tutors provide real-time feedback on student projects, allowing professors to act more like high-level mentors and coaches rather than graders.
 - **Simulation-Based Learning:** For complex fields like networking or cybersecurity, students use AI-powered simulations to test "What If" scenarios (e.g., "Simulate a 20% power grid failure") which were previously too expensive or dangerous to teach.
-

2. Core Skills for the Post-2026 Career

To be an "IT Engineer" now, you need a different toolkit:

- **Context Engineering:** This is "Prompt Engineering" on steroids. It's the ability to provide an AI with the correct domain data, constraints, and business logic so it produces consistent, production-ready results.
 - **AI Auditing & Governance:** Knowing how to "interrogate" an AI's output for bias, security holes, and hallucinations.
 - **System Architecture:** The ability to see the "big picture." Since AI builds the parts, the human must design the assembly line that connects those parts.
 - **Data Literacy & RAG:** Understanding how to structure and "clean" proprietary data so AI agents can use it effectively through Retrieval-Augmented Generation (RAG).
-

3. The "Ideal Student" Profile

In the past, IT was for people who could sit in a dark room and debug code for 10 hours. In 2026, the successful student looks different:

| | |
|-----------------------------------|---|
| Type of Student | Why they thrive in 2026 |
| The Polymath | Students who combine IT with another field (Biology, Law, Finance). AI handles the code; the human provides the specialized domain knowledge . |
| The Strategic Communicator | Since "coding" is now mostly "talking to a machine," those who can explain complex ideas clearly and logically are the best programmers. |
| The Ethical Auditor | Students with a natural skepticism. They don't trust the AI's first answer and enjoy hunting for the "why" behind a failure. |
| The Relentless Adapter | Because AI tools change weekly, students must have a "growth mindset" rather than a desire to master one static tool for 40 years. |
| Export to Sheets | |

Summary: Is this career right for you?

If you enjoy "puzzle-solving" at a high level and are comfortable being a **manager of machines** rather than a **manual laborer of code**, the 2026 IT landscape is incredibly rewarding. You will build more, faster, than any generation of engineers before you.

AND

Here are the most significant certifications and modules currently trending for IT students and engineers:

1. Professional Certifications for the "Agentic" Era

These certifications are highly valued in 2026 because they prove you can manage autonomous systems, not just write scripts.

- **NVIDIA Agentic AI LLMs Professional:** This is one of the most sought-after credentials for engineers. It focuses on "multi-agent interaction," distributed reasoning, and building "memory" into AI systems.
- **IBM RAG and Agentic AI (via Coursera):** A practical, project-based certification that teaches you how to build **Retrieval-Augmented Generation (RAG)** pipelines—essential for making AI "know" a company's private data.
- **ISACA Advanced in AI Audit (AAIA):** This has emerged as the "Gold Standard" for senior IT engineers and auditors. It focuses on identifying algorithmic bias, ensuring transparency, and protecting data privacy in large-scale AI deployments.

2. Trending University Modules (Late 2026)

If you are looking at university programs, these are the "new essentials" that have replaced traditional coding classes:

| Module Name | Core Focus | Why it's trending |
|-----------------------------------|--|---|
| Agentic Orchestration | Designing loops where AI agents plan, execute, and self-correct. | Replaces "Software Engineering I." |
| Formal Verification for AI | Using math to prove an AI-generated system is safe. | Crucial for "critical systems" (health, finance). |

| Module Name | Core Focus | Why it's trending |
|---------------------------------------|--|---|
| AI Governance & the EU Act | The legal and ethical framework for deploying AI in 2026. | Vital for compliance and safety roles. |
| Muti-Modal Data Pipelines | How to feed AI text, audio, video, and 3D data simultaneously. | Necessary for the next generation of "seeing" AI. |

3. Strategic Study Paths

Depending on your interest, you should follow one of these three paths:

- **The Builder (AI Engineer):** Focus on **Python, PyTorch**, and **LangChain/LangGraph**. Get certified in a cloud-specific platform (like *Azure AI Engineer* or *Google Professional ML Engineer*).
- **The Guardian (Security & Audit):** Focus on **AAIA (ISACA)** and **AI Quality (TÜV SÜD)**. Your job is to be the "skeptic" who ensures the AI isn't hallucinating vulnerabilities.
- **The Strategist (Product & Lead):** Focus on **AI Product Strategy** and **AI-Enabled Project Management**. These roles use AI to build entire businesses from scratch.

Summary of the "2026 Shift"

The most important thing to realize is that "**Vibe Coding**" (briefly describing what you want and letting the AI build it) is now a professional skill. However, to get the high-paying jobs, you must be the one who can **audit, secure, and scale** those vibes into reliable, multi-million dollar systems.