The Rise of the AI Agents

The Rise of the Coding Agents

AI-First Software Development

Fast-track tour of the AI revolution—supercharge coding with AI agents, then build products with AI Agents at the core of SaaS using reusable intelligence

Zia Khan

https://www.linkedin.com/in/ziaukhan/

Text Book: https://ai-native.panaversity.org/

The Rise of the AI-Driven Dev

The Rise of the Spec-Driven Dev

Zia Khan

Agentic AI Architect

Leadership

- CEO of Panaversity
- COO of PIAIC

Impact & Credentials

- Trained hundreds of thousands
- MBA, MSE, MAC from ASU
- CMA and CPA credentials (USA)

Nation-transforming AI education leader

ChatGPT Has Changed The World

Now Learn to Program in English or Urdu instead of Python or TypeScript

Your Journey to Becoming a Certified Agentic AI & Robotics Engineer

Transform from Code Writer to AI Orchestrator. This certification programme provides a progressive learning path from foundational AI development to advanced humanoid robotics.

LEVEL I: THE FOUNDATION
AI-DRIVEN & SPEC-ORIVEN DEVELOPMENT

Master AI-Driven Development
Learn to collakorate with AI coding agents like Claude, Gemini, and GPT-5.1.

Think in Specifications
Use Sosc-Driven Development (SDD) to create executable Sloeprints for generating code and tests.

Build with Modern Python
Master arwnc*ynuous and type-safe Python tailored for managing complex agentic workflows.

LEVEL II: BUILDING INTELLIGENCE
- AI ABENT & APPLICATION DEVELOPMENT

Construct Autonomous Agents
Build agents that can plan, remember (using RAG), and use tools to interact.

Integrate Complex Data Layers
Combine Vector, Relational, and Graph databases for comprehensive agent memory and knowledge.

Architect End-to-End AI Applications
Secure and containerise full-stack, multi-agent systems ready for deployment.

LEVEL III: SCALING SYSTEMS
- CLOUD-NATIVE & EVENT-DRIVEN AI

Deploy at Planetary Scale
Use Kubornetes and Depr to orchestrate hondreds of thousands of agents in the cloud.

Enable Asynchronous Communication
Implement event-driven architectures with Apache Kafka for high-throughput, decoupled systems.

Master Production Operations
Learn AIDps for managing observability, security, and CUCO pipelines for agentic systems.

LEVEL IV: ENTERING THE PHYSICAL WORLD - HUMANOID ROBOTICS

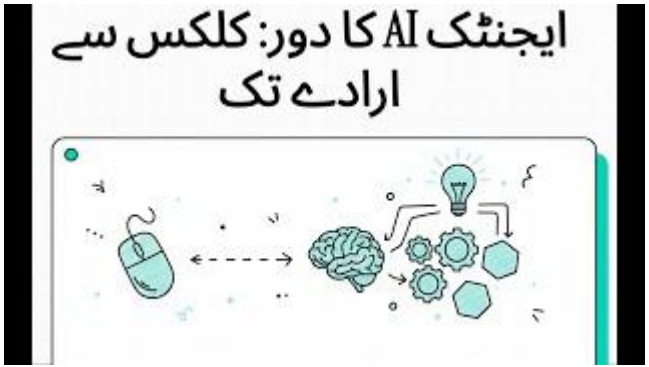
Bridge the Digital-Physical Divide
Give your AI agents a physical body and the ability to operate in reality.

Command the Rebotics Stack
Use RGS 2 for control, and Gazebo K WVIDIA laase for advanced simulation.

Create Conversational Robots
Integrate LLMs to build humanoids that can see, speak, and act on commands.

https://docs.google.com/document/d/1Rn-D3qIdMjJL-_HB4mE0tb5nCpxqlSUkwmiRFCcjiKc/edit?usp=sharing

Overview Videos in English and Urdu/Hindi

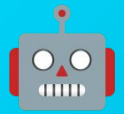


The Future of Work: A Partnership

Three forces working together

People

Judgment, creativity, oversight



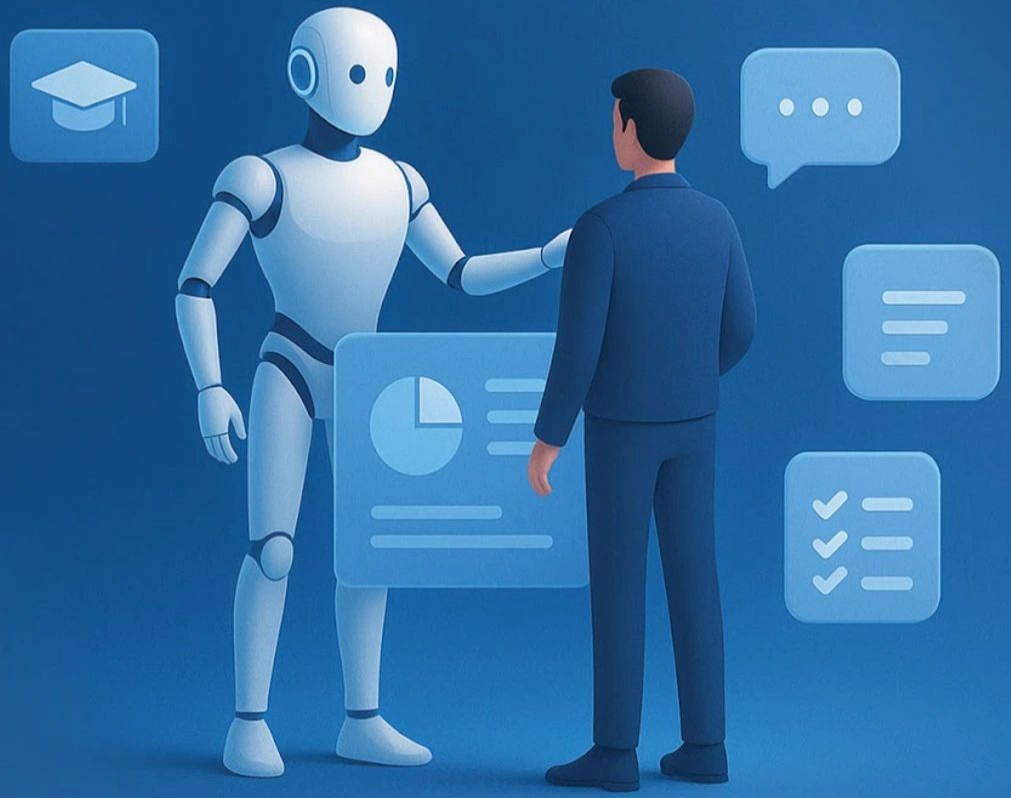
Agents

Digital work automation

Robots

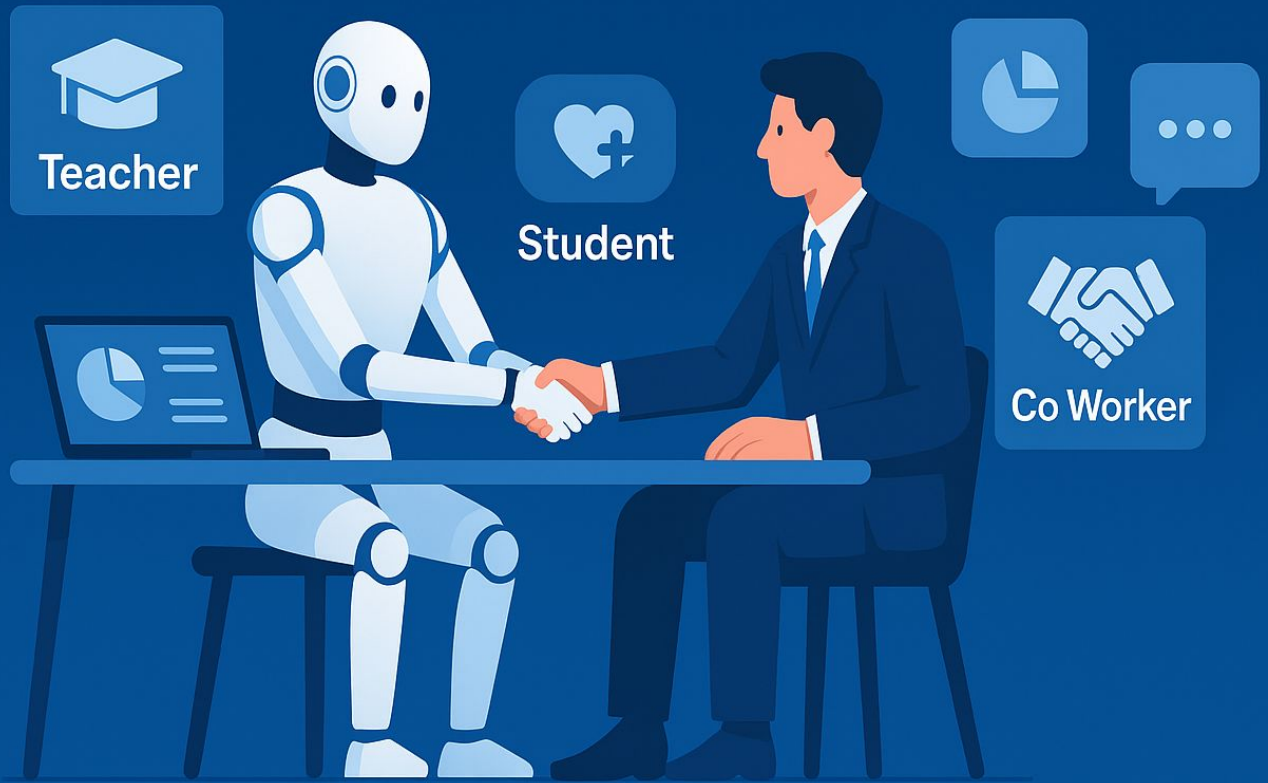
Physical work automation

This is How I See My Life From Now On



AI is my Teacher, my Student, my Co Worker

Together, we will do everything.



The Three Waves of AI

Predictive AI

Focus on analyzing data to predict outcomes.

Analyze past data to predict future outcomes.

Why It Mattered: Enabled data-driven decision-making.

Generative AI

Focus on creating content from data.

Creating content (text, images, code, videos).

Why It Mattered: Empowered creativity and productivity.

Agentic AI

Focus on autonomous actions and learning iteratively.

Autonomous actions, environment interaction, iterative learning.

Why It Matters: AI becomes proactive, managing complex tasks.

What is an AI Agent?

An AI agent is a piece of software that can pursue a goal by observing its environment, deciding what to do next, taking actions (often by calling tools/APIs or controlling a robot), and learning from the results—then repeating the loop until the goal is met.

What makes it an “agent” (not just a chatbot)?

Goal-driven: You give it an objective (“pull the Xero trial balance daily and export CSV”), not just a single prompt.

Tool use / actions: It can call functions, APIs, databases, browsers, or devices—not only generate text.

State & memory: It keeps context (short-term working state and longer-term memory) across steps.

Autonomy: It plans multi-step work, executes, checks results, and adjusts without you micromanaging.

From User Interface to User Intent

The Age of Agentic AI is Here

ChatGPT: The World's First True UI for AI

The Linguistic Interface

Conversation, not clicks

No Language Barrier

Human-computer interaction happens through natural conversation

Beyond Chat

Moving past traditional chat windows into something fundamentally new

From **Python/Java/TypeScript…** to **natural-language–first**

The Next Leap in AI

Moving from understanding to action

Large Language Models

AI that responds

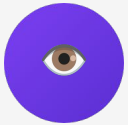
→

Large Action Models

AI that acts, orchestrates, and remembers

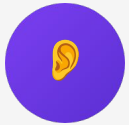
Autonomous Agents: The Five Powers

Systems that can see, hear, reason, act, and remember



See

Visual understanding



Hear

Audio processing

Reason

Complex decision-making

Act

Execute and orchestrate

Remember

Maintain context and learn

The Paradigm Shift

From UX to Agentic Experience

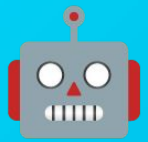
Traditional UX

User initiates every action

Manual navigation through interfaces

Clicks and commands

Agentic Experience



AI anticipates and acts

Autonomous orchestration

Intent and conversation

Redefining Everything

How agentic AI transforms our world

How We Work

AI agents as collaborative teammates

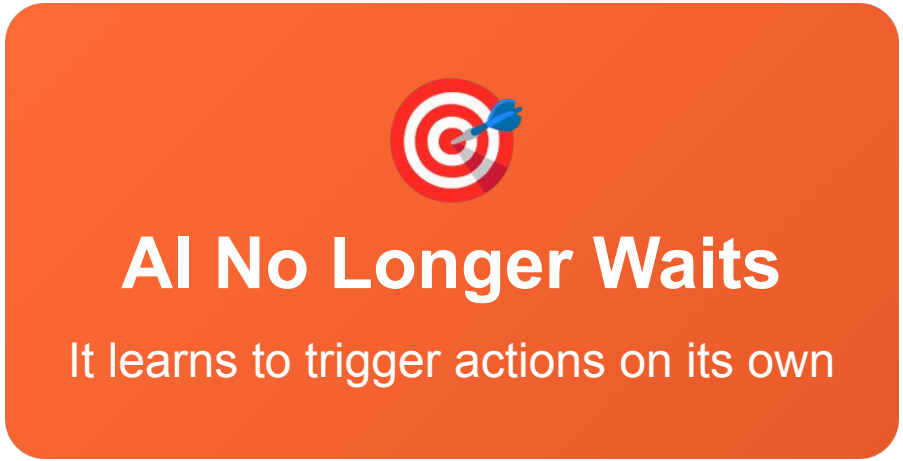
How We Transact

Autonomous systems managing complex transactions

How We Build

AI-first development from high-level intent

The Future: Proactive AI



"We're moving from large language models to large action models where AI doesn't just respond, it acts, orchestrates, and remembers."

— Sandeep Alur

The Agentic AI Era

Key characteristics of the new paradigm

1

Autonomous

Acts independently without constant human input

2

Contextual

Understands and adapts to situations

3

Orchestrative

Coordinates complex multi-step workflows

4

Persistent

Remembers and learns from interactions

The Future is Agentic

AI is no longer waiting for instructions

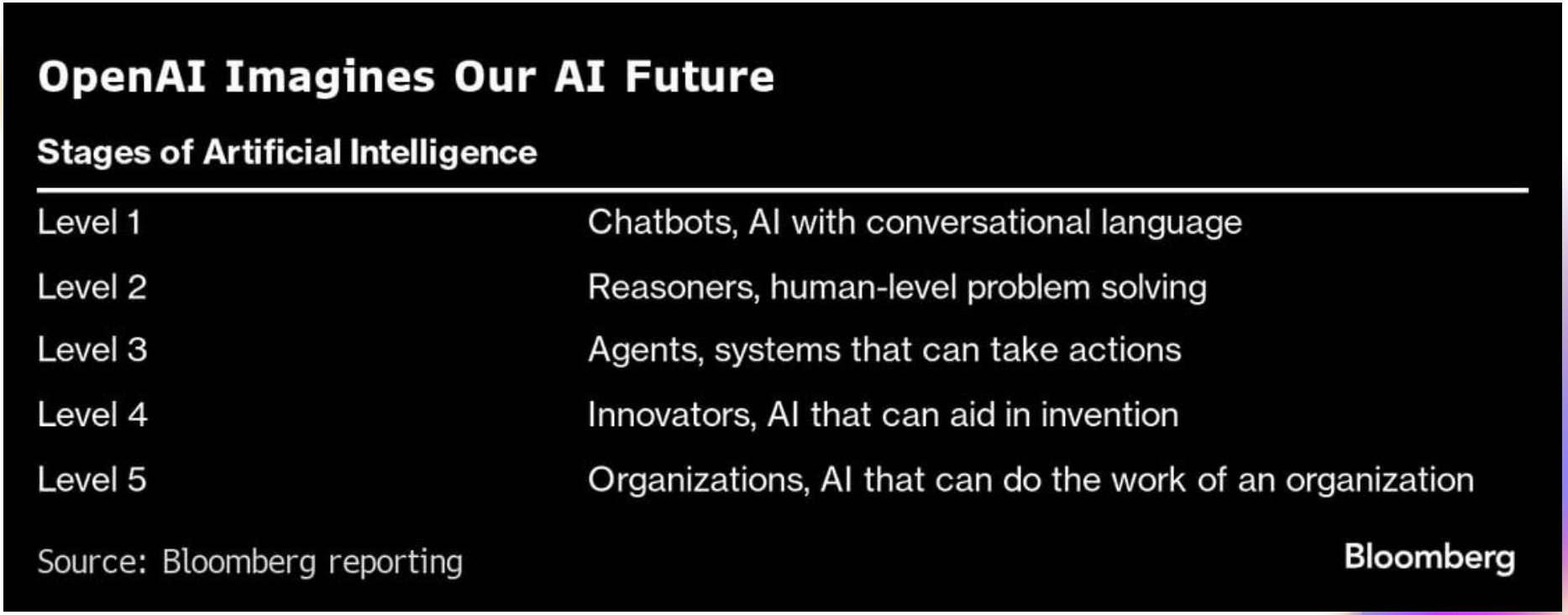
From user interface to user intent

From responding to orchestrating

From passive to proactive

Embrace the agentic revolution

The Road to AGI



OpenAI Imagines Our AI Future

Stages of Artificial Intelligence

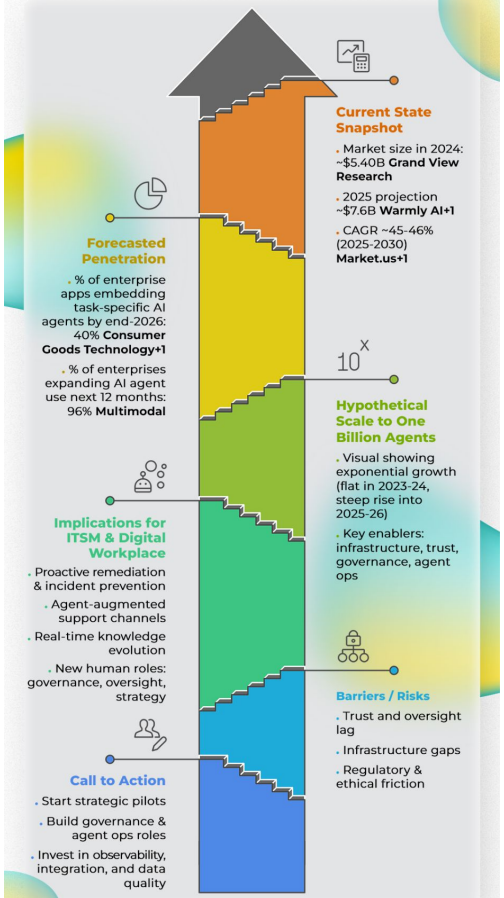
Level 1	Chatbots, AI with conversational language
Level 2	Reasoners, human-level problem solving
Level 3	Agents, systems that can take actions
Level 4	Innovators, AI that can aid in invention
Level 5	Organizations, AI that can do the work of an organization

Source: Bloomberg reporting

Bloomberg

One Billion AI Agents by 2026

What It Could Mean for Your ITSM & Digital Workplace



Current State Snapshot

- Market size in 2024: ~$5.40B **Grand View Research**
- 2025 projection ~$7.6B **Warmly AI+1**
- CAGR ~45-46% (2025-2030) **Market.us+1**

Forecasted Penetration

- % of enterprise apps embedding task-specific AI agents by end-2026: 40% **Consumer Goods Technology+1**
- % of enterprises expanding AI agent use next 12 months: 96% **Multimodal**

Hypothetical Scale to One Billion Agents

- Visual showing exponential growth (flat in 2023-24, steep rise into 2025-26)
- Key enablers: infrastructure, trust, governance, agent ops

Implications for ITSM & Digital Workplace

- Proactive remediation & incident prevention
- Agent-augmented support channels
- Real-time knowledge evolution
- New human roles: governance, oversight, strategy

Barriers / Risks

- Trust and oversight lag
- Infrastructure gaps
- Regulatory & ethical friction

Call to Action

- Start strategic pilots
- Build governance & agent ops roles
- Invest in observability, integration, and data quality

REZOLVE.ai


Humans and AI Agents
Collaborating for a bright future
AI
AI
AI

The AI Revolution in Software

Two parallel paths transforming how we build technology

AI-Driven

Supercharge Coding

Use AI as your powerful development partner

AI-Native

Build for AI

Create products with AI at the core

The AI Revolution in Software

Three parallel paths transforming how we build and deploy technology

Use Coding Agent to Program (Claude Code)

Supercharge Coding

Develop AI Agents (Open Agents SDK)

Create Products with AI at the core

AIOps (Using AI Agents for AI Operations)

Maintain, Monitor, & Scale AI Systems

The AI Development Revolution

The most significant transformation in software development

The scale of transformation

The **$3 trillion developer economy** (equivalent to France's GDP) and why it's being restructured in 2-3 years instead of the typical 10-15 year cycle.

Developer evolving role

The shift from **developer-as-typist** to **developer-as-orchestrator.**

Why this is different?

Internal disruption (**software disrupting itself**), universal impact (all roles affected), unprecedented speed, and the recursion effect.

The autonomous agent era

The Evolution from code completion → function generation → feature implementation → autonomous agents (Gen 1 through Gen 4).

The shift from **developer-as-typist** to **developer-as-orchestrator**

The Philosophy: Co-Learning Between Human and Machine

What Makes This Different

Traditional education: "Instruct the computer what to do"

AI-native era: "Learn together" — humans and agents refining each other's understanding

In this model:

- **You explain** what you want (in a specification)
- **AI suggests** how it might be done (generating code)
- **You evaluate** the output and learn from it
- **AI learns** from your feedback and refines
- **Together** you converge on a working solution

This feedback loop — **co-learning** — is the heart of AI-native development. It's not about replacing the developer; it's about *augmenting* your reasoning, creativity, and speed.

Your Evolving Role: Teacher + Student + Orchestrator

In the AI-native world, you blend three identities:

Teacher: Guiding the AI's understanding of purpose through clear specs

Student: Learning new patterns, architectures, and techniques from AI suggestions

Orchestrator: Designing how humans, AIs, and agents collaborate to solve problems

You're no longer just writing code — you're conducting an orchestra of intelligences.

The Spec-Driven Way: From Intent to Implementation

Specifications as Living Contracts

A specification is no longer static documentation. It's a **living contract** between you and your AI collaborator.

AI-Driven Development: The Complete Workflow

AIDD is an end-to-end process:

Specification — You describe what should exist (the contract)

Generation — AI drafts scaffolds, routes, components (rapid execution)

Execution — Test, deploy, monitor (automated validation)

Reflection — Agents analyze results and improve (continuous learning)

This is recursive: Better specs → Better code → Better data → Smarter AI → Better specs

That's the feedback loop that powers co-learning.

Thinking Like an AI-Native Developer

The Mindset Shift: From Logic to Language

Old paradigm: Tell computers *exactly* what to do (write syntax)

New paradigm: Tell them *roughly what you mean* (write intent)

The syntax no longer matters as much as the **intent**.

Your success depends on how well you can describe problems, constraints, and goals to intelligent systems.

In other words: Specs are the new syntax.

AI-Driven Development

Using AI to supercharge coding

What It Is

AI coding agents as assistants to write, debug, and refactor code faster

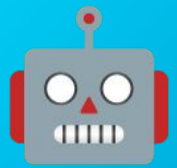
Key Tools

- Claude Code - Agentic Development
- Gemini CLI - Open Source
- Zed - AI-native Code Editor

10x faster development

AI Native Development

Building products with AI at the core



AI as

Product Core

What It Is

Designing systems where AI agents and models are fundamental components

Examples

- Autonomous AI agents handling tasks
- Self-learning recommendation systems
- AI-powered customer service

Future is AI-first architecture

The Convergence

Best teams leverage both approaches simultaneously

AI-Driven

Build faster with AI tools

+

AI Native

Build AI-powered products

Use AI to build AI products at unprecedented speed

Mindsets for Success

Mental models that separate AI leaders from laggards

Prompt Engineering

Master the art of communicating with AI effectively

Spec-Driven

Clear specifications lead to better AI outputs

Rapid Iteration

Test, learn, improve at unprecedented speed

AI-First Thinking

Design with AI capabilities in mind from day one

Essential Tools Ecosystem

The modern AI development toolkit

Coding Agents

- Claude Code
- Gemini CLI
- OpenAI GPT5-Codex
- Zed IDE (AI Native IDE)

Spec-Driven Development

- Panavesity Spec-Kit Plus
- Amazon Kiro
- Wessl

AI Frameworks

- OpenAI Agents SDK
- Model Context Protocol
- Anthropic Agents SDK

Deployment

- Vercel / Netlify
- Docker / Kubernetes / Dapr
- Ray


PROGRAM IN ENGLISH
CLAUDE CODE
def create website(;
user-input = listen('What kind ɗ website?)
website-template
" generate code(uɔsuut)
deploy(website))
GEMINI CLI
$ gemini run "build mobile app"
"android, iOS"
$ gemini deploy
"gemini integrate
"user database"
PROGRAM IN URDU

Claude Code

Agentic coding from terminal

What It Does

Delegates coding tasks, creates files, runs tests

Key Features

- Autonomous execution
- File management
- Command execution
- Sub Agents
- Skills

Delegate Tasks,

Not Just Code

Gemini CLI

Google terminal AI assistant

Multimodal

Power

What It Does

Multimodal AI for code generation and analysis

Key Features

- Image and code understanding
- Natural language queries
- Fast response times
- Free Tier (1,000 Requests Per Day)

Spec-Driven Development - the future of building digital products

Spec-Driven Development

Clear specifications unlock AI agent potential

The Problem with "Vibe Coding"

Unclear requirements lead to endless iterations and unpredictable outputs from AI agents

The Solution

Write detailed specs before coding. AI agents execute better with clear instructions.

Benefits

- Consistent AI outputs
- Fewer iterations needed
- Better team alignment

Spec First → AI Executes → Quality Results

Spec Kit Plus

Structured specs for AI agents

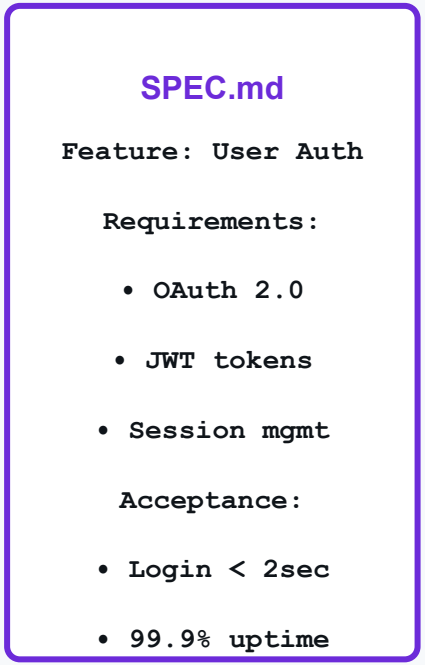
What It Provides

Templates and standards for writing clear, actionable specifications that AI agents can execute

Core Components

- Feature specifications
- Vertical Sub agents and Skills
- Prompt History and Architecture decision records
- Test Driven Development

Transform ideas into AI-executable specs



The Evolution of Development Artifacts

Two parallel transformations reshaping software development

Code-Centric

Source code as the primary asset

Spec-Driven

Specifications as the primary asset

The Traditional Code-Centric Model

Decades of organizing around human-authored source code

- Source code as the canonical representation of system behavior
- Shaped version control, architecture patterns, and career paths
- Reusability meant crafting libraries and frameworks
- Evolved from procedural to object-oriented to functional paradigms
- Yet the assumption persisted: code is the primary asset

The Emergence of Generative Development

AI capabilities challenging code-centric assumptions

- On-demand code generation from natural language descriptions
- Automated refactoring preserving behavior
- Continuous synchronization between requirements and implementation
- Multi-framework migration with reduced manual intervention

The bottleneck moves from writing code to expressing intent with precision.

A New Abstraction Layer

The historical evolution of programming

Machine code → Assembly

Made instructions human-readable

Assembly → High-level languages

Abstracted away hardware details

High-level code → Specifications + AI

Abstracts away implementation patterns

Languages like Python and TypeScript become intermediate representations.

Redefining Reusability for the AI Era

From reusable code to reusable intelligence

Legacy

Modular libraries, design patterns, components

Commoditization

AI excels at boilerplate, CRUD, tests

Reusable Intelligence

Specialized agents, domain skills, orchestration patterns

Anatomy of AI Coding Agents

Core components of effective agents and sub-agents

Persona and Behavioral Profile

Well-defined identity: testing specialists, performance engineers, security auditors

Domain-Specific Skills

Bundled knowledge and tools for healthcare compliance, financial processing, monitoring

Tool Access via MCP

Model Context Protocol integration for unified access to development environments

Spec-Driven Development with Reusable Intelligence

The new paradigm for software development

The Durable Asset

Specifications and agent architectures become primary

Specs + RI Define:

- System behavior and constraints
- Agent personas and responsibilities
- Skills, MCP servers, orchestration

The Panaversity Teaching Method

Four-layer framework for AI-native software engineering education

Teaching AI Native technologies using AI-Driven and Spec-Driven Development through systematic progression from foundational understanding to full spec-driven project execution.

Layer 1

Manual Practice

Layer 2

AI-Assisted

Layer 3

Intelligence Design

Layer 4

Spec-Driven Projects

Layer 1: Foundation Through Manual Practice

Applied to each lesson

Purpose

Establishes conceptual understanding independent of AI tools

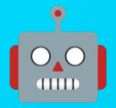
The Approach:

- Step-by-step walkthroughs
- Manual CLI operations
- Hand-written code examples
- Explaining purpose

Students understand before directing AI

Layer 2: AI-Assisted Execution

Applied to each lesson



Purpose

Translates manual workflows into AI-assisted workflows

Learning Outcomes:

- Express tasks through natural language prompts
- Use coding agents to generate implementations
- Debug agent outputs effectively
- Analyze trade-offs in AI suggestions

AI tools as collaborative partners

Layer 3: Designing Reusable Intelligence

Applied to each lesson

Purpose

Transforms lesson knowledge into reusable agent components

Activities:

- Define specialized subagents
- Create skills bundling instructions and tools
- Configure for reuse across projects
- Document usage patterns

Creating reusable intelligence assets

Layer 4: Spec-Driven Project Integration

Applied once per chapter

Purpose

Integrates chapter knowledge through projects

Project Activities:

- Design using spec-first approaches
- Use Spec-Kit Plus for structure
- Compose subagents and skills
- Orchestrate multi-agent workflows

Reusable intelligence compounds

Progressive Complexity

Building fluency across all four layers

Students develop fluency in:

Manual Execution

Foundational understanding

AI-Augmented

Collaborative workflows

Intelligence Architecture

Component design

Spec-Driven Delivery

Full project execution

Each lesson builds individual concepts while chapters integrate multiple concepts through real-world project complexity.

Implications for Developers

Strategic shifts in competencies and career paths

Core Competencies

Specification design and systems thinking alongside traditional programming

Emerging Career Paths

AI Systems Designer, Intelligence Engineer, Spec Architect, Agent Orchestrator

Continuous Learning

MCP integration, agent patterns, specification frameworks, evaluation methods

Evolve from code author to intelligence architect

Implications for Organizations

Strategic shifts in engineering practices

Investment Priorities

Shift from one-off code assets to curating specification and intelligence libraries

Engineering Practices

Version control for specifications, CI/CD with spec validation, intent-focused documentation

Quality Assurance

Specification-driven test generation and automated compliance through specialized agents

Code Review Evolution

Emphasis on alignment with specifications, not just code quality

Implications for Education

Curriculum modernization requirements

Modernization Needs:

- AI-native development from the beginning
- Specification design integrated throughout
- Practical experience with real AI tools

Assessment Approaches:

- Evaluate specification quality
- Assess agent system design ability

Challenges and Limitations

Not all domains suit spec-driven approaches equally

Domain Limitations:

- Novel algorithms require traditional methods
- Performance-critical systems need optimization
- Legacy maintenance may not benefit

Organizational Challenges:

Transitioning codebases, varying team skills, specification quality, agent orchestration complexity

The Structural Evolution

Code's role is shifting

Old Constraint

How fast can we write code?

New Constraint

How well can we design systems and make decisions?

The spec-driven paradigm posits that specifications capture intent with precision, agent architectures define specialized roles, reusable intelligence becomes the strategic differentiator, and code is generated and regenerated as specs evolve.

Conclusion for Developers

A gradual but profound transition

Teams and individuals who thrive will think in terms of:

- Reusable intelligence — not just reusable code
- Precise specifications — guiding AI-powered implementation
- Orchestrated collaboration — between humans and AI agents

This parallels historical transitions: from machine code to assembly, assembly to high-level languages, and now from manual coding to specification-driven generation.

1

Ideate and Define Specs

Define problem and desired solution with clear specs

2

Prompt

Use AI agents to generate code, architecture, tests

3

Iterate

Rapidly refine with AI feedback loops

Ideas to working systems in hours, not weeks

Spec Driven Development

The Nine Pillars

1. **AI CLI & Coding Agents** (tools like Claude Code, Gemini CLI, OpenAI GPT5-Codex)
2. **Markdown as Programming Language** (natural language specifications become executable)
3. **MCP Standard** (Model Context Protocol—universal tool integration)
4. **AI-First IDEs** (editors like Zed and Cursor built for AI collaboration)
5. **Linux Universal Dev Environment** (standardized development through WSL/Mac/Linux)
6. **Test-Driven Development** (TDD for quality confidence at scale)
7. **Specification-Driven Development with SpecKit Plus** (structured methodology)
8. **Composable Vertical Skills** (reusable domain expertise components)
9. **Universal Cloud-Native Deployment** (standardized infrastructure with Kubernetes, Docker, Dapr)

The Dual Language Stack: Python + TypeScript

Every AI system lives between two worlds

Python: The Reasoning World

TypeScript: The Interaction World

Real-World Impact

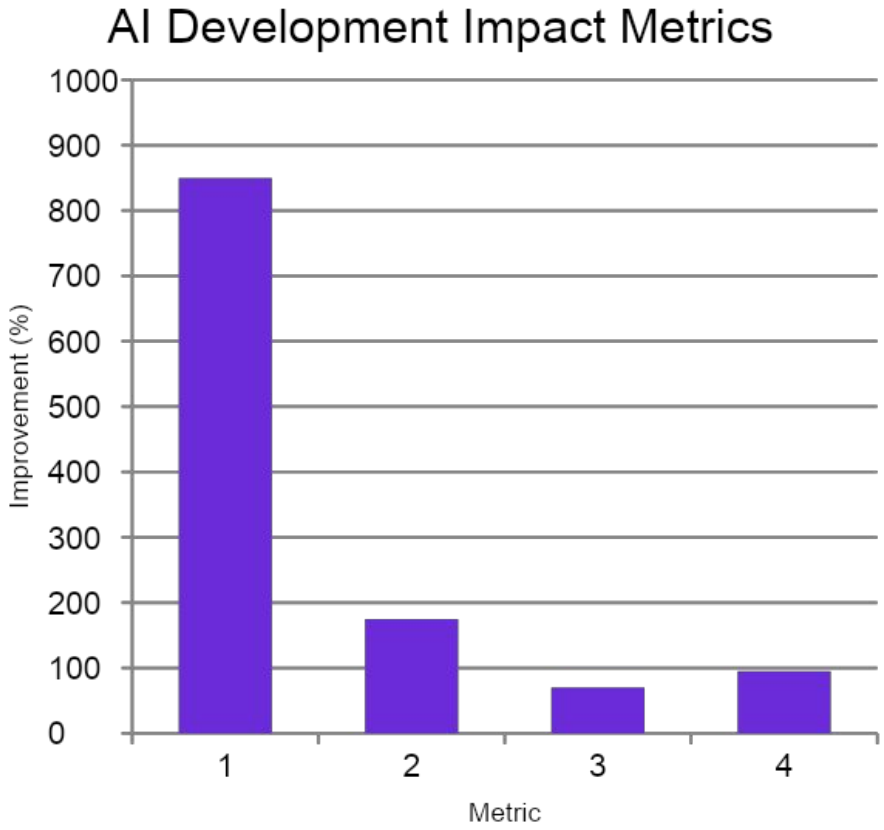
Quantifying the revolution

10x

Faster Development

70%

Time to Market Cut



Your Roadmap to Success

Concrete steps to start your AI journey

Start with AI coding tools

Try Claude Code, Gemini CLI, or Cursor

Master Prompt Engineering and Spec-Driven Development

Learn to communicate with AI agents

Build AI-augmented projects

Apply AI-Driven and AI Native concepts

Explore AI Native architectures

Learn OpenAI Agents SDK, Google ADK, Anthropic Agents SDK, MCP

Join the community at Panaversity

The Question You're Asking

You're probably asking one of these questions:

"Am I too late?"

Beginner wondering if AI has closed the window for new developers.

"How do I teach this?"

Educator unsure how to prepare students for this landscape.

"Will this replace me?"

Experienced developer concerned about career security.

"Is this real or hype?"

Skeptic wanting evidence before investing time and energy.

The answer to all four is the same, and it might surprise you:

This is the best time in decades to be learning software development.

Not despite AI. **Because of it**.

This is the best time in decades to be learning software development

The barriers that kept people out of programming for fifty years—memorizing syntax, debugging cryptic error messages, understanding compiler optimization, configuring development environments—are dissolving. AI tools handle these mechanical tasks while you focus on what actually matters: **understanding problems, designing solutions, and building systems that create value.**

But there's a catch. The skills that traditional computer science education emphasizes—algorithm memorization, syntax fluency, low-level implementation details—are exactly the skills AI tools are best at automating. If you're learning to code the way universities taught it in 2020, you're preparing for a job that's already obsolete.

The Paradox: Developers Are More Valuable, Not Less

Here's what surprises people:

As AI tools become more powerful, skilled developers become MORE valuable, not less.

Why? Because the constraints shift:

Old constraint: How fast can we write code?

New constraint: How quickly can we design good systems and make correct decisions?

The Market is Expanding

When code generation was slow (human typing speed), that was the bottleneck. Now the bottleneck is:

- Understanding what to build
- Designing architectures that scale
- Making trade-off decisions
- Ensuring quality and security
- Coordinating across systems

All of these require human expertise, judgment, and creativity.

Additionally, because AI tools make developers more productive, the **demand for software is increasing**, not decreasing. Companies that previously couldn't afford custom software can now build it. Individuals can create tools for personal use. The market is expanding.

Why Traditional CS Education Falls Short

Let's address the uncomfortable truth: **The computer science education you can get at most universities in 2025 is teaching you how to be a developer in 2015.**

Traditional CS education still has tremendous value.

Universities operate on a **2-4 year curriculum revision cycle**.

Minimum time: 2 years. **Typical time:** 3-4 years.

Meanwhile, the AI coding landscape is evolving on a **3-6 month cycle.**

The result: By the time a university updates curriculum to address an emerging technology, that technology has already evolved past what's being taught.

How to Make a Billion Dollars in the AI Era?

Solo developers and tiny teams are building billion-dollar businesses

The Snakes & Ladders framework

Competing in **vertical markets** (healthcare, legal, logistics) offers better odds than competing at the consumer layer.

Vertical intelligence paradigm

The shift from building reusable code libraries to building **reusable AI-driven intelligence systems** that understand domain-specific workflows, compliance, and nuances.

Super orchestrator economics

tiny teams generate billion-dollar value by **orchestrating AI** to handle the mechanical 90% while humans focus on the creative 10%.

The Piggyback Protocol Pivot strategy

Three-phase playbook for entering vertical markets—start with broad AI tools (Piggyback), build vertical reputation and relationships (Protocol), pivot to your chosen domain with credibility (Pivot).

Solopreneur

Could Agentic AI Create a One-Person Unicorn?

The AI revolution has already minted dozens of unicorns—startups valued at $1 billion before going public. Now it will create a whole new type of startup: **The One-Person Unicorn**

Start Learning Now: Open Source Book

PANAVERSITY AI-NATIVE BOOK SERIES

AI Native Software Development

Colearning Agentic AI with Python and TypeScript – The AI & Spec Driven Way

✨ Open Source | 🤝 Co-Learning with AI | 🎯 Spec-Driven Development

Start Reading → | Explore Panaversity

https://ai-native.panaversity.org

Learn and Implement Everything:

The AI Drive Way

Bash, Git, and Github

Using Claude Code and Gemini CLI

Cloud-Native: Kubernetes, Docker, Dapr, Ray

Using kubectl-ai, kagent (kagent-dev), Claude Code, and Gemini CLI

Python: The Language of Agents

Using Claude Code and Gemini CLI

TypeScript: The Language of Interaction

Using Claude Code and Gemini CLI

Combo Pattern for AI-Native Development

OpenAI Agents SDK is the main Orchestrator. It keeps each framework doing what it's best at, without lock-in. Almost any agent/RAG/workflow framework can slot into this pattern.

OpenAI Agents SDK

Orchestrate with OpenAI Agents SDK (handoffs + tracing).

The two universal ways to plug anything in are: Expose it as an HTTP microservice and call it from the OpenAI Agents SDK as a function tool (like the ADK/Claude agents). Or wrap it as an MCP server and attach it as an MCP tool.

Google Agent Development Kit (ADK)

For Gemini-specific tasks or GCP data ops, expose ADK agents behind HTTP/MCP/A2A and call them as tools from your OpenAI orchestrator.

Claude Agent SDK

For developer-automation jobs, compose Claude Agent SDK sub-agents (computer use/code tools) and surface them via MCP or HTTP. Call them as tools from OpenAI Orchestrator with function calling/MCP.

Model Context Protocol (MCP)

Plug in MCP servers for your internal tools and data. All framework support MCP making interop easy. You can also wrap any Agentic Framework as an MCP server and attach it as an MCP tool.

Rethink Everything

‘We’re suddenly in a moment where it’s time to rethink everything’

Agentic AI is the Future:

The Agentic Web

Open Protocols & Closed Ecosystems

Building the Internet of AI Agents

The Vision: Internet of AI Agents

AI agents communicating, collaborating, and transacting autonomously

From Human-to-AI to Agent-to-Agent

AI agents will discover, negotiate with, and collaborate with other agents without human intervention

Discoverable

Agents find each other

Interoperable

Standard protocols

Autonomous

Self-directed action

Two Paths Emerging

Open protocols vs closed ecosystems

Open Web

Decentralized, interoperable agents on open protocols

Closed Gardens

Platform-controlled ecosystems with proprietary SDKs

Both approaches shaping the future of AI agents

Agent-to-Agent Protocol

Direct communication standard

What It Enables

Standardized way for AI agents to discover, authenticate, and communicate

Core Capabilities

- Agent discovery and registry
- Secure authentication
- Message exchange formats
- Task delegation protocols

Agents Talk to Agents

Open Web Agents

Decentralized AI ecosystem

Key Principles

- No single point of control
- Open standards and protocols
- Permissionless innovation
- User data sovereignty

Benefits

- Cross-platform interoperability
- Developer freedom
- Competition and innovation

Example Scenario

Travel agent discovers and coordinates with booking, weather, and translation agents across different platforms seamlessly

Like email: any agent can reach any other agent

NANDA

The Internet of AI Agents

Network of Agentic Nodes for Distributed Automation

Open framework enabling autonomous AI agents to discover, communicate, and collaborate across a decentralized network

Discovery

Trust

Messaging

Orchestration

NANDA Architecture

Core components powering the agent internet

Agent Registry

Decentralized directory of available agents, capabilities, and endpoints

Identity & Authentication

Cryptographic identity system ensuring secure agent-to-agent communication

Message Protocol

Standard format for requests, responses, and event notifications

Workflow Engine

Coordinates multi-agent tasks and manages complex interactions

Built on open standards for maximum interoperability

Walled Garden

Approach

OpenAI Apps

ChatGPT ecosystem

The Model

Platform-controlled app ecosystem within ChatGPT

Key Features

- Curated app marketplace
- Integrated user experience
- Platform revenue sharing

ChatGPT Apps Ecosystem

Features and capabilities

Advantages

- Large built-in user base
- Simplified development
- Platform handles hosting
- Discoverability in store

Trade-offs

- Platform lock-in
- Limited cross-platform use
- Revenue sharing required

Example Apps

- Productivity tools
- Data analysis apps
- Creative assistants
- Domain-specific agents

Like iOS: controlled but powerful

OpenAI Apps SDK

Building for the platform

What Developers Get

Tools to build apps that integrate with ChatGPT interface

SDK Components

- Authentication APIs
- UI component library
- State management

app.yaml

```
name: MyApp
version: 1.0
capabilities:
  - chat
  - web_search
permissions:
  - user_data
```

ChatGPT App

Open vs Closed: The Tradeoffs

Each approach has strengths

Open Web

Wins:

- Maximum flexibility
- No platform tax
- True interoperability

Challenges:

- More complex setup
- Discovery is harder

Closed Garden

Wins:

- Built-in user base
- Easier development
- Better discovery

Challenges:

- Platform dependency
- Revenue sharing

The Future: Convergence

Best of both worlds emerging

Open Standards

\+

Platform Value

Hybrid Models Emerging

- Platforms adopting open protocols for interoperability
- Open frameworks providing platform-like discovery
- Bridges connecting closed and open ecosystems

Agents will flow freely between worlds

Building for Both Worlds

Strategic approach for developers

Start with open standards

Build core agent logic using open protocols like NANDA and A2A

Create platform adapters

Build thin wrappers for ChatGPT, Claude, and other platforms

Enable cross-platform discovery

Register in both open registries and platform stores

Maintain agent portability

Keep business logic separate from platform-specific code

Maximize reach while maintaining flexibility

Agentic AI is the Future:

The Agentic Organization

The Next Paradigm for the AI Era

Reimagining Enterprise for Human-AI Collaboration

The Largest Organizational Shift

Since the industrial and digital revolutions

Humans + AI Agents

Working side by side at scale at near-zero marginal cost

89%

Organizations still in industrial age

9%

Using digital-age models

1%

Operating as agentic networks

Exponential AI Capability Growth

The acceleration of autonomous work

Task Length Doubled Every 4 Months Since 2024

Current: AI can complete ~2 hours of work. Projection: 4 days by 2027

Intern Level

Constant supervision

Mid-Level

Independent work

Executive

Strategic direction

Technology & Data

Infrastructure for the agentic era

Agentic AI Mesh

Democratized technology with universal integration architecture

Agent Protocols

Easier integration across systems

Build vs Buy

Based on competitive advantage

The Organizational Evolution

From hierarchies to agentic networks

Industrial Age

Hierarchical structures

89%

Digital Age

Agile teams

9%

Agentic Era

Flat networks

1%

Enormous Competitive Advantage

Early adopters will capture disproportionate value

Three Radical Shifts

Leadership imperatives for transformation

1. Linear to Exponential

Adapt operating models boldly - replace silos with autonomous teams

2. Technology-Forward to Future-Back

Start with future vision when agents run 60% of operations

3. Threat to Opportunity

Reframe AI as opportunity for human capability extension

How to Start: The Roadmap

Think boldly, move fast, go deep

1. Top Team Agenda

Agentic AI in leadership discussions

2. CEO Vision

Vision for agentic organization

3. AI Center of Excellence

Ramp up capabilities

4. Upskill People

Continuous reinvention

5. Lighthouse Domains

Launch agentic processes quickly and learn live

Key Takeaways

Critical insights for leaders

⚡

Speed Matters

Adapt faster to capture value

👥

Human-Centric

Humans orchestrate & lead

🏗️

Holistic Transform

All pillars evolve together

🎯

Start Now

Act, learn, and adapt

Agentic AI is the Future:

Agentic Commerce

The AI Revolution in Shopping

How AI Agents Are Transforming Consumer and Merchant Experiences

What is Agentic Commerce?

Shopping powered by AI agents acting on our behalf

A Seismic Shift in the Marketplace

AI agents anticipate needs, navigate options, negotiate deals, and execute transactions autonomously

Autonomous

AI acts independently via multistep reasoning

Intent-Driven

Aligned with human goals

Frictionless

Fast, integrated flow

The Massive Market Opportunity

Projected growth by 2030

$1T

US B2C Retail

Orchestrated revenue

$3-5T

Global Market

Total opportunity

Faster Than Web or Mobile

Agents ride existing digital infrastructure instead of waiting for new rails to be built

Three Key Interaction Models

How agentic commerce takes shape

1. Agent to Site

Agents interact directly with merchant platforms (e.g., scanning hotel websites)

2. Agent to Agent

Agents transact autonomously with other agents (e.g., negotiating bundle discounts)

3. Brokered Agent to Site

Intermediary systems facilitate multiagent interactions (e.g., OpenTable broker)

Key Protocols Enabling Agentic Commerce

The infrastructure standards

ACP

OpenAI + Stripe commerce protocol

AP2

Google payments protocol

A2A

Agent-to-agent communication

MCP

Model Context Protocol

Benefits for Consumers

A transformed shopping experience

Personalized

Offers aligned with budget and preferences

Frictionless

No navigation, fewer clicks

Intelligent

AI compares options and negotiates

Convenient

Automated recurring orders

Benefits for Merchants

New opportunities for growth

Reach High-Intent Buyers

Sell through AI agents using existing commerce infrastructure

Dynamic Pricing

Real-time optimization

Reduced Friction

Fewer abandoned carts

The Agentic Commerce Ecosystem

Key players and infrastructure

AI Platforms

OpenAI, Google, Anthropic

Payments

Stripe, Visa, Mastercard

Merchants

Shopify, Etsy, retailers

Interconnected Network

Similar to e-commerce ecosystem but designed for autonomous agent transactions

Challenges and Risks

Navigating the new landscape

Overspending Risk

Fewer steps = fewer abandoned carts

Trust & Security

Verify agent authenticity

Disintermediation

Risk to traditional platforms

Compliance

KYC/AML for agents

What Merchants Need

Becoming agent-ready

Agent-Ready APIs

Clear interfaces for agent interactions

Optimized Product Data

Structured for agent discovery

Rethink Identity & Loyalty

Delegated access and agent authentication

Minimal Infrastructure Changes

Works with existing payment and commerce systems

The Protocol Battle

OpenAI vs Google approaches

OpenAI ACP

Product-led, speed to market with ChatGPT Instant Checkout

First Mover

Live with Etsy and Shopify merchants

Google AP2

Consortium approach with 60+ partners

Open Standard

Broad industry adoption

Disrupting Traditional Gatekeepers

Shifting power dynamics

From Search Engines to AI Agents

Discovery moves from Google/Amazon to conversational agents

4,700%

Year-over-year traffic increase from GenAI to retail sites

New Power Brokers

AI platforms control discovery and fees

How to Prepare for Agentic Commerce

Action steps for merchants and platforms

1. Assess Readiness

Evaluate current systems

2. Choose Protocols

ACP, AP2, or both

3. Optimize Data

Make products discoverable

4. Test & Learn

Start with pilot programs

Agents, Robots, and Us

Skill Partnerships in the Age of AI

At a Glance

Key findings on the future of work

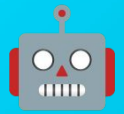
- 57% of US work hours could be automated with today's technologies
- 70% of current skills will remain relevant but be applied differently
- AI fluency demand has grown 7x in two years—fastest of any skill
- $2.9 trillion in economic value could be unlocked by 2030
- Work will be a partnership between people, agents, and robots

The Future of Work: A Partnership

Three forces working together

People

Judgment, creativity, oversight



Agents

Digital work automation

Robots

Physical work automation

57% of Work Hours Could Be Automated

But this doesn't mean job losses—it means work transformation

44%

By agents (nonphysical work)

13%

By robots (physical work)

People remain vital for oversight, quality control, and tasks requiring empathy, creativity, and judgment. Many jobs will change as specific tasks are automated, shifting what people do rather than eliminating work.

Most Skills Will Remain Relevant

But how they're applied will evolve

72%

of skills are used in both automatable and non-automatable work

Less Time On:

Preparing documents, basic research, routine tasks

More Time On:

Framing questions, interpreting results, making decisions

AI Fluency: The Fastest-Growing Skill

Demand has exploded across all industries

7x

Growth in demand over two years

AI fluency has grown faster than any other skill. Roles from SEO specialists to chemists to financial managers now require this capability.

The Skill Change Index

Measuring automation's impact on skills

Most Exposed

Digital and information-processing skills (accounting, specific programming languages)

Evolving Skills

Problem-solving, communication, quality assurance—changing in application rather than demand

Least Exposed

Interpersonal skills (coaching, negotiation), assisting and caring skills

Nearly every occupation will experience skill shifts by 2030

Seven Work Archetypes

How people, agents, and robots collaborate

People-Centric (33%)

Healthcare, maintenance—largely human

Agent-Centric (35%)

Legal, admin—high digital automation

Robot-Centric (5%)

Drivers, operators—physical automation

People-Agent (20%)

Teachers, engineers, finance specialists

People-Robot (1%)

Maintenance, construction—tools enhance

People-Agent-Robot (5%)

Transport, agriculture, food service

Agent-Robot (2%)

Manufacturing—software directs systems

$2.9 Trillion in Economic Value by 2030

If organizations prepare people and redesign workflows

Annual US Economic Value

$2.9T

Success depends on redesigning workflows and adapting human skills rather than new technological breakthroughs.

Reimagining Workflows Is Key

Beyond automating individual tasks

Why workflows matter:

- 60% of productivity gains are in sector-specific functions
- Applying AI to legacy processes delivers limited gains
- Full redesign enables people, agents, and robots to work together

Nearly 90% of companies invested in AI, but fewer than 40% report measurable gains—often because they automate tasks instead of redesigning workflows.

Case Study: AI-Powered Sales Process

A global tech company transformed its sales workflow

AI Agents

Prioritization, outreach, response management, scheduling, handoff

Results:

- 7-12% revenue increase
- 40-50% time saved
- Focus on relationships

Humans shifted to strategic engagement and partnership building

Case Study: Customer Operations

A utility company handling 7 million support calls annually

Agentic AI System

Authentication, intent ID, scheduling, self-service—40% of calls

Results:

- 50% cost reduction per call
- +6 points customer satisfaction
- Handle complex issues

Representatives focus on emotionally complex issues, agents resolve routine inquiries

Case Study: IT Modernization

A regional bank modernizing its SME banking application

AI Agent Factory

Assessment, functionality, coding agents—15-20 agents per developer

Results:

- 70% code accuracy
- 50% reduction in hours
- Shift to orchestration

Human work shifted to planning, orchestration, and quality verification

Key Questions for Business Leaders

Critical decisions for AI transformation

Are you reimagining for future value?

Redesigning entire processes vs. improving existing workflows

Are you leading as a transformation?

Senior leadership commitment vs. IT delegation

Are you building a culture of experimentation?

Supporting curiosity, risk-taking, and learning from setbacks

Key Questions for Leaders (continued)

People, trust, and skills

Are you building trust and ensuring safety?

Clear policies, validation, exception handling, human oversight

Are you equipping managers for hybrid teams?

Leading people, agents, and robots—coaching and orchestration

Are you preparing workers for new skills?

Skill-based pathways, continuous learning, AI fluency across all levels

Key Questions for Institutions

Preparing society for AI transformation

How can education keep pace?

AI fluency from primary school, transferable skills, continuous learning systems

How do transferable skills lead to opportunities?

Clear skill definitions, trusted credentials, better matching platforms

How can local economies respond?

Data-driven understanding, collaboration between educators, workforce agencies, and employers

The Transformation Ahead

What success requires

Redesign Workflows

Not just automate tasks—reimagine entire processes

Invest in People

Skills, training, and pathways for the future

The outcomes for firms, workers, and communities depend on how organizations and institutions work together to prepare people for the jobs of the future.

Core Insights

Key takeaways from the research

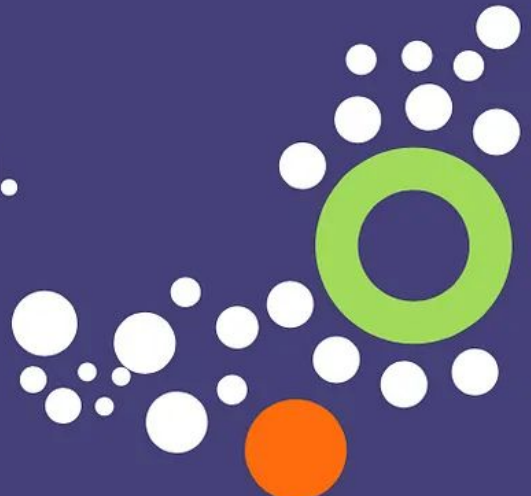
- Work will be a partnership of people, agents, and robots
- Automation transforms work, doesn't eliminate it
- Most human skills remain relevant but evolve
- AI fluency is the fastest-growing workforce requirement
- Workflow redesign unlocks greater value than task automation

Investing in workers and their skills—not just technology—will be decisive in ensuring AI's benefits are widely shared

There comes a time
we need to stop reading the
books of others.

And write our own.

- Albert Einstein



Join the Spec-Driven Revolution

Learn AI-native development with reusable intelligence at Panaversity

Panaversity.org | PIAIC.org | Connect Today