

# AI Driven & AI Native Development

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

---

**Zia Khan**

CEO Panaversity • COO PIAIC

# Zia Khan

## Agentic AI Architect

### Leadership

- CEO of Panaversity
- COO of PIAIC

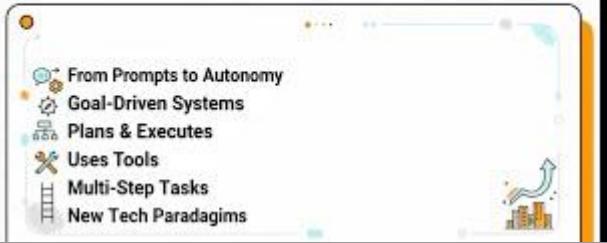
### Impact & Credentials

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

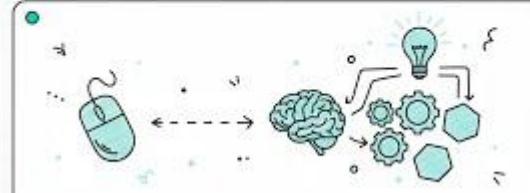
Nation-transforming AI education leader

# Overview Videos in English and Urdu/Hindi

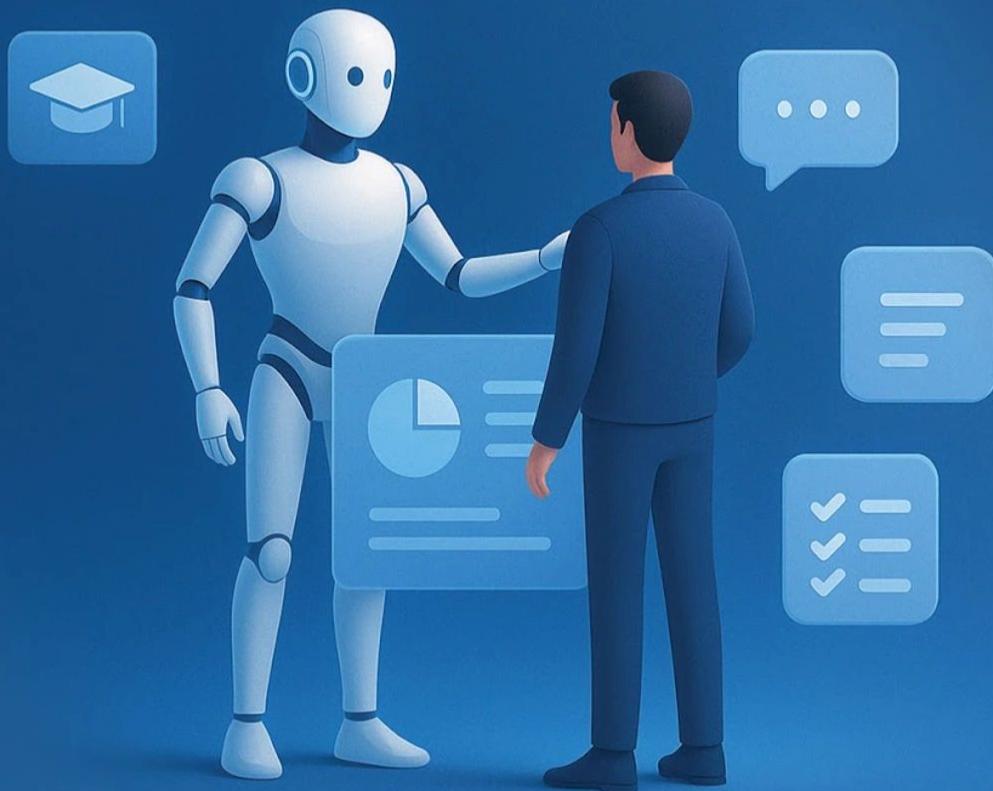
## Agentic AI: The Revolution



## ایجنتک AI کا دور: کلکس سے ارادہ تک

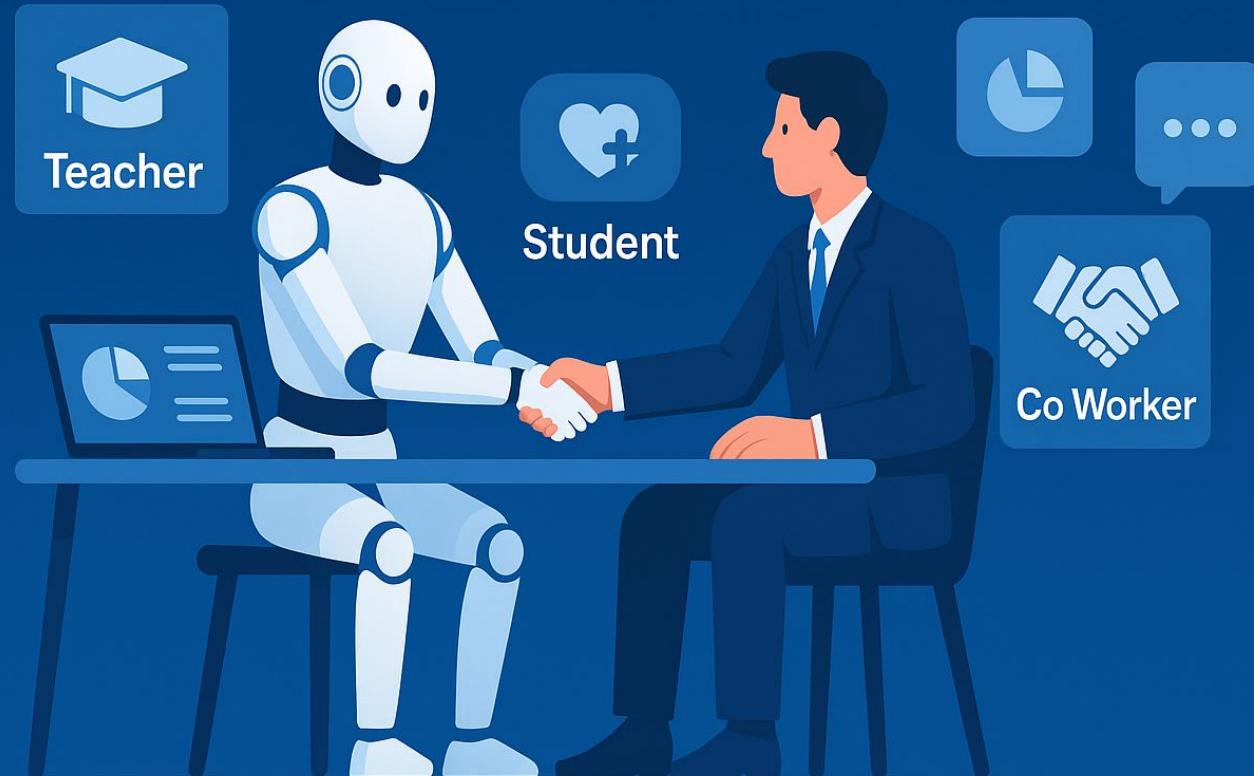


# This is How I See My Life From Now On



# 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

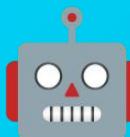


User initiates every action

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



## AI No Longer Waits

It learns to trigger actions on its own

*"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

## Autonomous

- 1 Acts independently without constant human input

## Contextual

- 2 Understands and adapts to situations

## Orchestrateive

- 3 Coordinates complex multi-step workflows

## Persistent

- 4 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

---

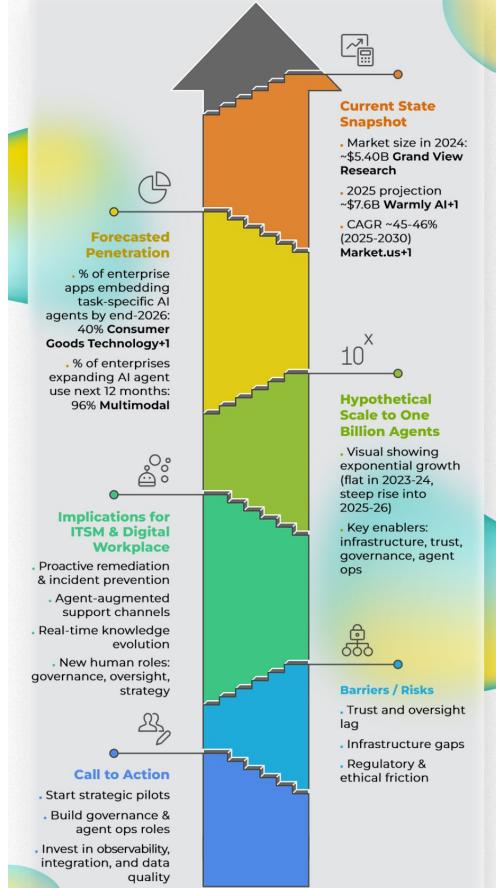
Level 1	Chatbots, AI with conversational language
Level 2	Reasoners, human-level problem solving
Level 3	Agents, systems that can take actions
Level 4	Innovators, AI that can aid in invention
Level 5	Organizations, AI that can do the work of an organization

Source: Bloomberg reporting

Bloomberg

# One Billion AI Agents by 2026

What It Could Mean for Your ITSM  
& Digital Workplace



# Humans and AI Agents

Collaborating for a bright future



# 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 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**.

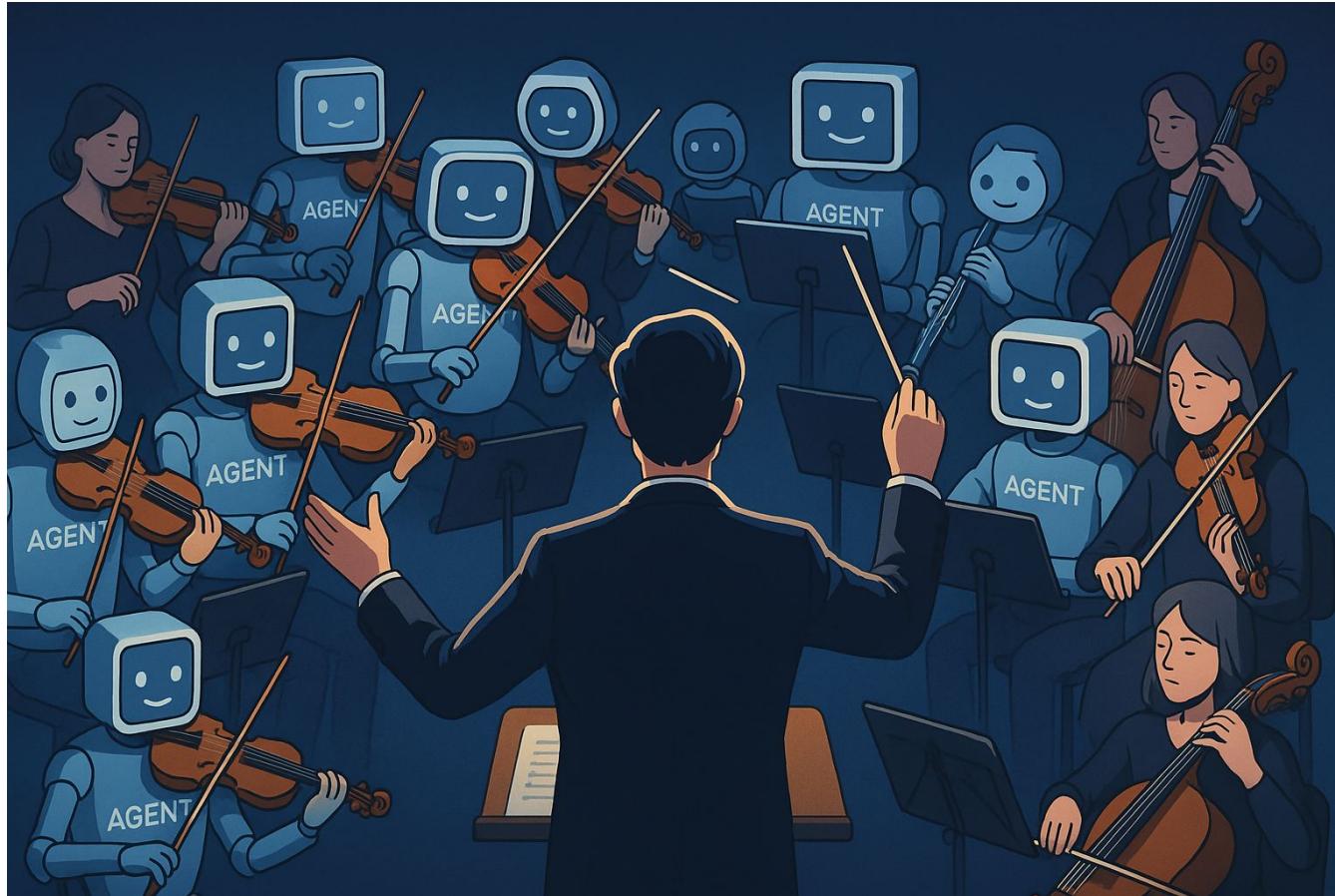
## **Why this is different?**

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

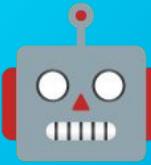


Code at

Warp Speed

# 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

# 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



Multimodal  
Power

# Gemini CLI

Google terminal AI assistant

## 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

99x

## Spec-Driven Development



# 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

SPEC.md

Feature: User Auth

Requirements:

- OAuth 2.0
- JWT tokens
- Session mgmt

Acceptance:

- Login < 2sec
- 99.9% uptime

Production Code

→ AI Agent →

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

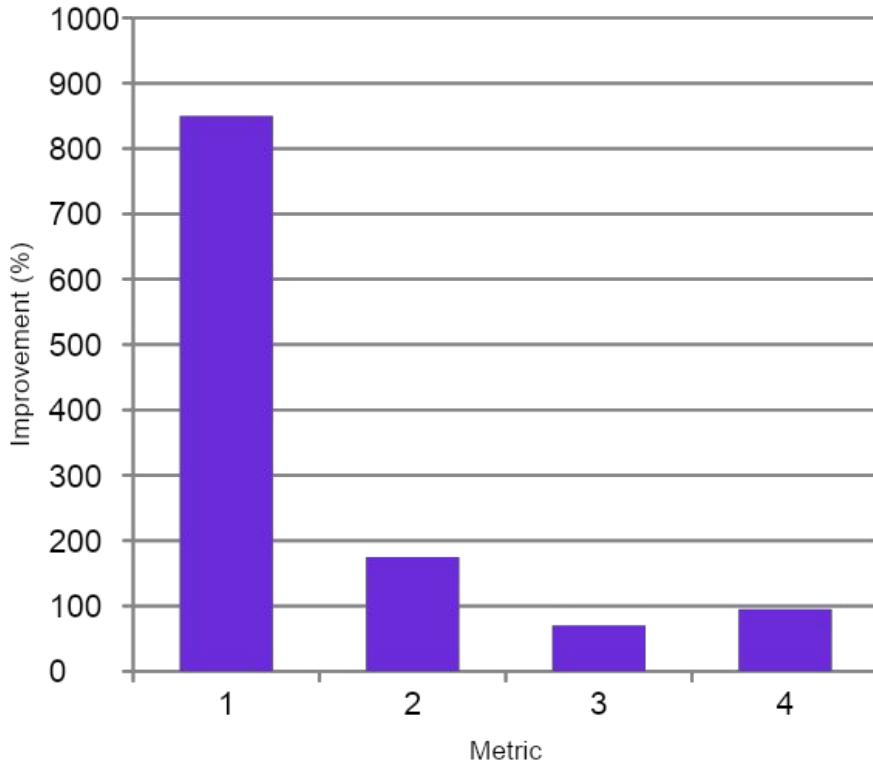
**10x**

Faster Development

**70%**

Time to Market Cut

AI Development Impact Metrics



# 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.

## "Will this replace me?"

Experienced developer concerned about career security.

## "How do I teach this?"

Educator unsure how to prepare students for this landscape.

## "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.

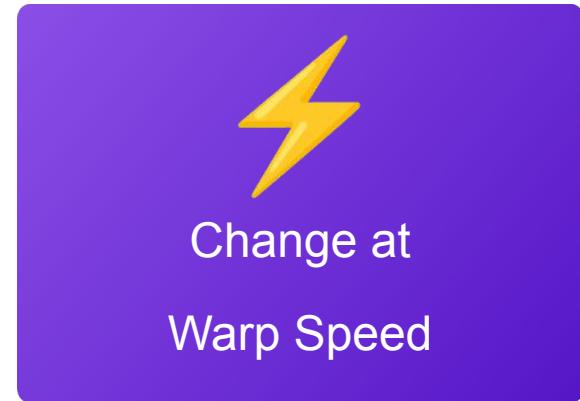
**Traditional CS education still has tremendous value.**

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).

# 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.



## 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.



## 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.



## 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

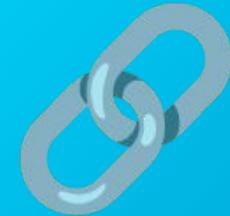
Direct communication standard

## 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

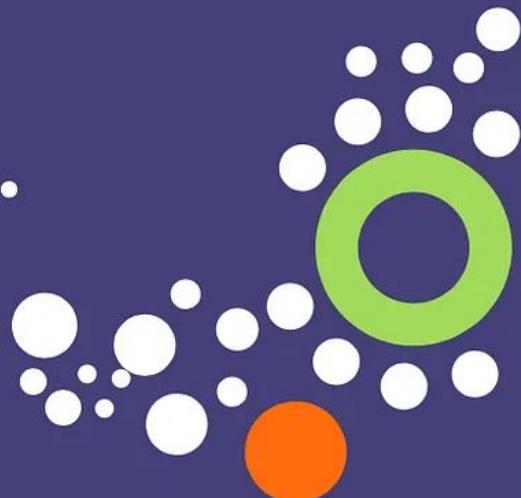
## 4. Test & Learn

Start with pilot programs

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

**And write our own.**

- Albert Einstein





# Start Your AI Journey Today

The AI revolution is happening now. Join thousands learning to build the future.

Panaversity.org

PIAIC.org

Connect Today

Transform ideas into reality at warp speed