

From Copilots to Autonomous Systems

What's actually changing? and what it means for builders?

Prasanna Jayapal

INTRODUCTION - *Prasanna Jayapal*

Co-Founder & CTO - AI Startup

Building a platform to help small businesses to harness the power of AI.



Previously: Meta, Teradata, Microsoft

Technical leader with 25+ years of industry experience and have served as Director and Head of Engineering in large scale tech companies.

Domains: Cloud, Data Platforms, Enterprise Management, ML Infrastructure, and AI systems.



Masters in CS
Leadership & Mgmt

Engineering Leader

Agenda Highlights

*We are moving from AI that answers to AI that executes workflows.
This workshop offers a glimpse into building real-world solutions.*

Tech Talk

- *Evolution of Agentic AI, its Adoption and Pitfalls*
- *Modern Agentic Architectures and Design Patterns*

Hands-on Workshop

- *Designing an Agentic AI Workflow*
- *When multi-agent is useful*

Career Growth & Strategies in the Agentic AI Era

Evolution of Agentic AI

How we got here?

1970s – 1990s



Traditional AI

Rule-Based Systems
If-then logic, expert systems,
hand-coded rules.

*Rigid, couldn't adapt or
learn from new data.*

1995 – 2010s



Predictive AI

Machine Learning
Learn patterns from data.
Classification, prediction, recs.

*Powerful but narrow
one model <-> one task.*

2019+



Generative AI

Generates text, code, images.
General-purpose &
conversational.

*Creates Content,
but cannot take actions*

Generative AI to Agentic AI

GenAI != Copilot

Gen AI generate responses. Copilots embed into real tools/workflows.

Copilot != Agent

Copilots assist humans. Agents can plan, act, and complete tasks.

Many “Agents” Today = Prompt + Tool call + Hope

The Reality: Evolution, Not Revolution

This shift didn't happen overnight.

It's the result of layering of capabilities over time.

The Beginning of Co-Pilots

Augmenting Human Decision-Making

Copilots provide real-time guidance and suggestions to enhance human decision-making and task execution.

Interactive Partnership

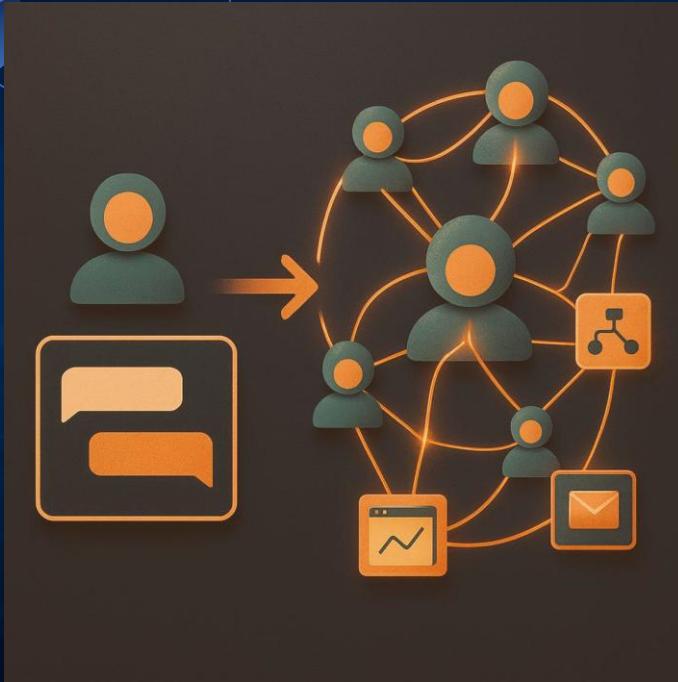
Copilots act as partners helping users navigate complex workflows without replacing human input.

Wide Range of Applications

Copilots are used in software like code completion and hardware such as pilot assistance in aviation.



From LLM Copilots to Agentic Systems



From Reactive to Proactive

Shift from prompt-following responses to agents that plan, reason, and take initiative.

Multi-Step Workflows

Agents decompose tasks, execute sequences, and verify outputs end-to-end.

Real-World Integration

APIs and tools enable external actions, customer support, data analysis, and more.

Key Technological Shifts Enabling Agentic AI



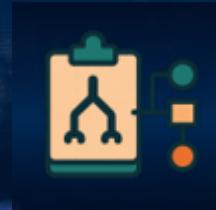
Advances in Language and Learning

Progress in natural language understanding and continual learning enables smarter and adaptable AI agents.



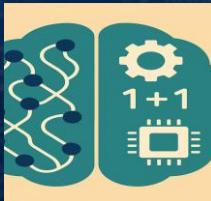
Innovations in Model Architecture

Transformer models with enhanced context and retrieval-augmented generation drive dynamic and modular AI task execution.



Improved Computing Power & Memory

Enhanced processing power and distributed computing support complex AI agents that plan multi-step actions.



Integration of Symbolic Reasoning

Combining symbolic reasoning with deep learning enhances AI's ability to represent knowledge and reason effectively.

Rise of Multi-Agent Thinking as a New Paradigm



Multi-Agent Paradigm

Specialized sub-agents collaborate or compete to solve complex problems effectively and flexibly.

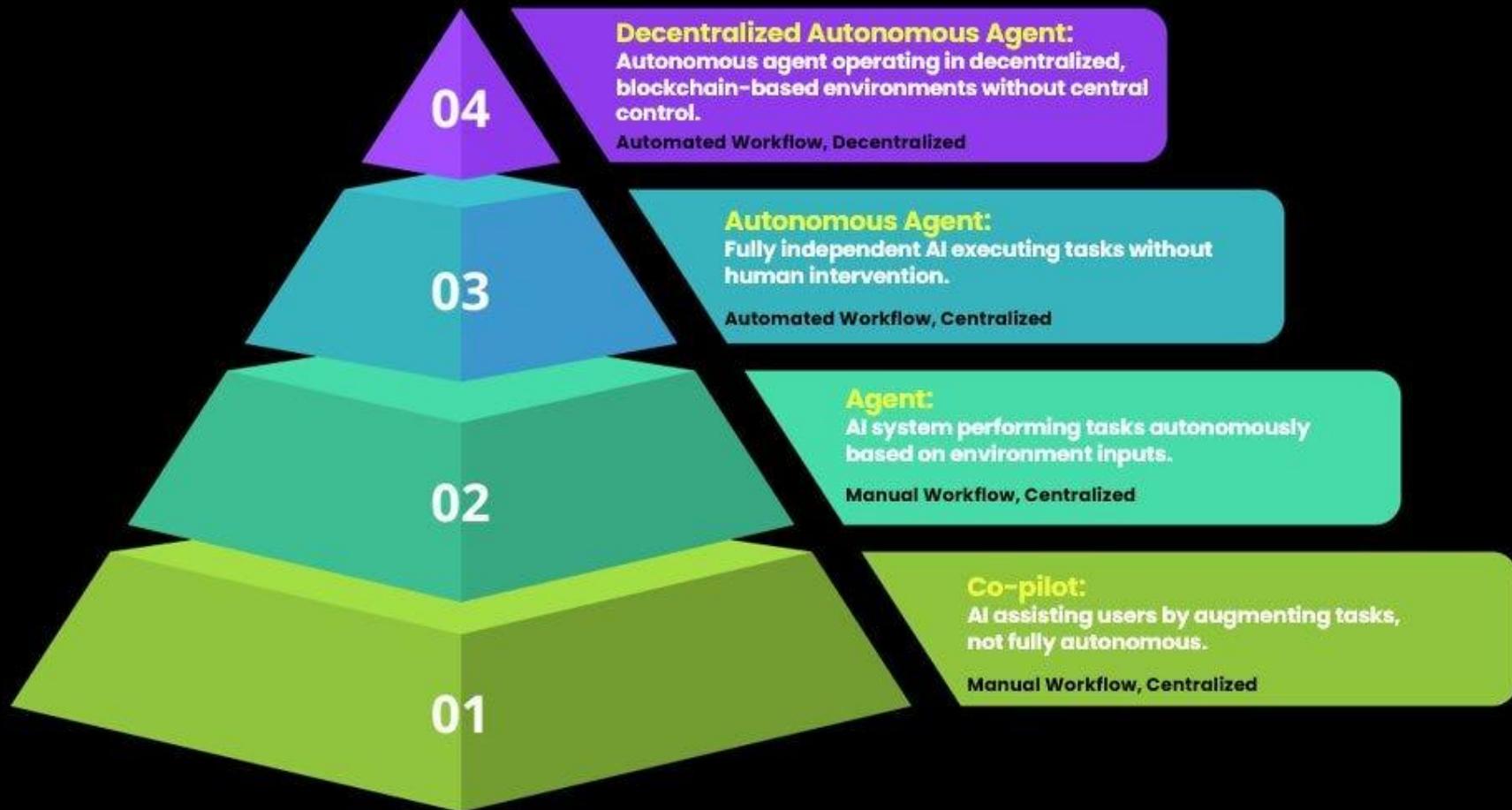
Inspired by Human Organizations

Mirrors Org structures, leveraging diverse skills.

Applications and Emergent Behavior

Cooperative problem solving & distributed decisions.

Agentic Maturity Model



Agentic AI in Real Industry: Adoption, Patterns, and Pitfalls

Where Agentic AI Is Actually Being Used

Adoption Is Happening in Bounded, High-Value Workflows.

Common Production Use Cases

- Engineering: code migration, test generation, release validation
- IT / Ops: incident triage, log analysis, root-cause workflows
- Enterprise Knowledge: contract review, RFP drafting, compliance evidence
- Data Teams: schema mapping, pipeline repair, metadata enrichment

Agents succeed where tasks are structured, repeatable, and tool-driven.

Enterprise and Product Use Cases



Operational Efficiency Gains

Early adopters report 30-40% reductions in manual workloads and improved customer satisfaction through AI adoption.

Common Pitfalls and Scaling Issues



- Treating agents like chatbots (stateless, no control loop)
- No observability and can't explain what the agent did
- Scaling and Coordination Overhead
- Ignoring evaluation and no measurable definition of “correct”
- Trust, Ethics, and Compliance
- Lack of guardrails, rollback, and deterministic fallbacks

Lessons Learned From Failed Implementations

Over-Ambitious Scope

Failures often arise from overestimating AI autonomy, poor data quality, and misaligned business goals.

Ignoring Humans & Cost Controls

Lack of escalation paths and cost/latency planning made solutions impractical to scale.

Best Practices Not Adopted

Incremental rollouts, thorough testing, and hybrid human-agent workflows improve project success rates.



Modern Agentic Architectures and Design Patterns

Agents operate in structured execution loops

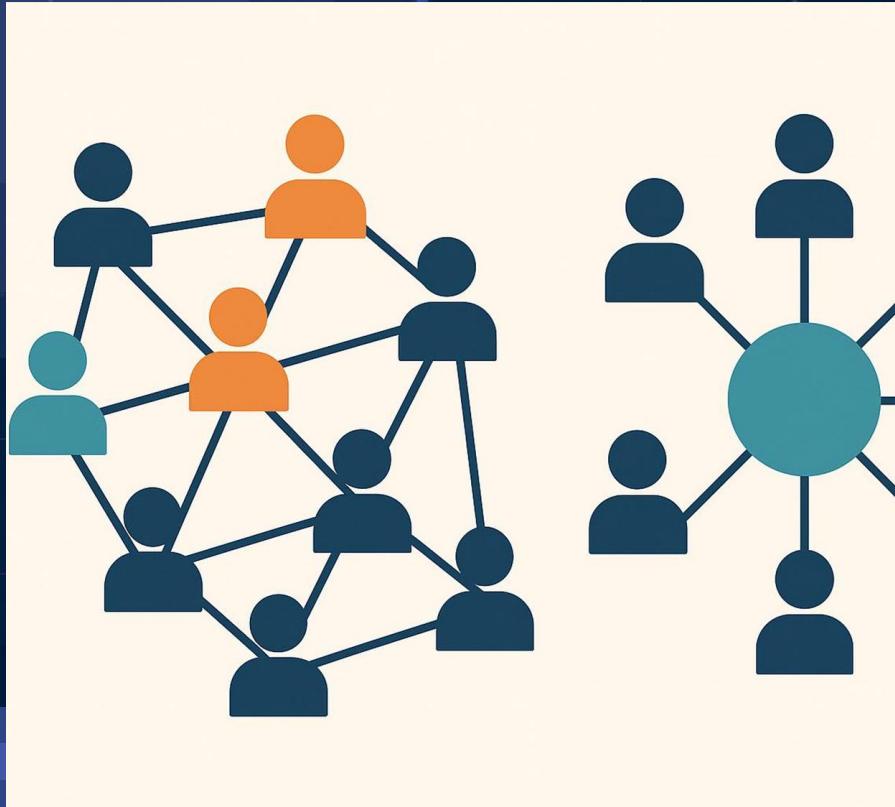
Its Not Just the Model

- Reasoning Layer: LLM
- Execution Layer: APIs
- Memory Layer: State
- Control Layer: Orchestration
- Trust Layer: Evaluation

Agentic Design Patterns Taking Shape

- **Goal-Driven Execution** — Systems translate intent into plans and execute toward outcomes, not just answers.
- **Closed-Loop Operation (Observe–Plan–Act–Verify)** — Iterative reasoning with validation makes agents reliable.
- **Tool-Centric Architecture** — LLM decides *what to do*; APIs, services, and code do the work.
- **Role-Based Multi-Agent Collaboration** — Specialized agents (planner, executor, validator) mirror real tam structures.
- **Stateful + Retrieval-Grounded Workflows** — Memory and dynamic context enable continuity and domain awareness.
- **Workflow-Orchestrated Autonomy** — Deterministic control layers add observability, guardrails, and cost management.

Agent Orchestration and Coordination Strategies



Coordination Models

Coordination strategies include centralized, decentralized peer-to-peer, and hybrid models balancing control and autonomy.

Collaboration Protocols

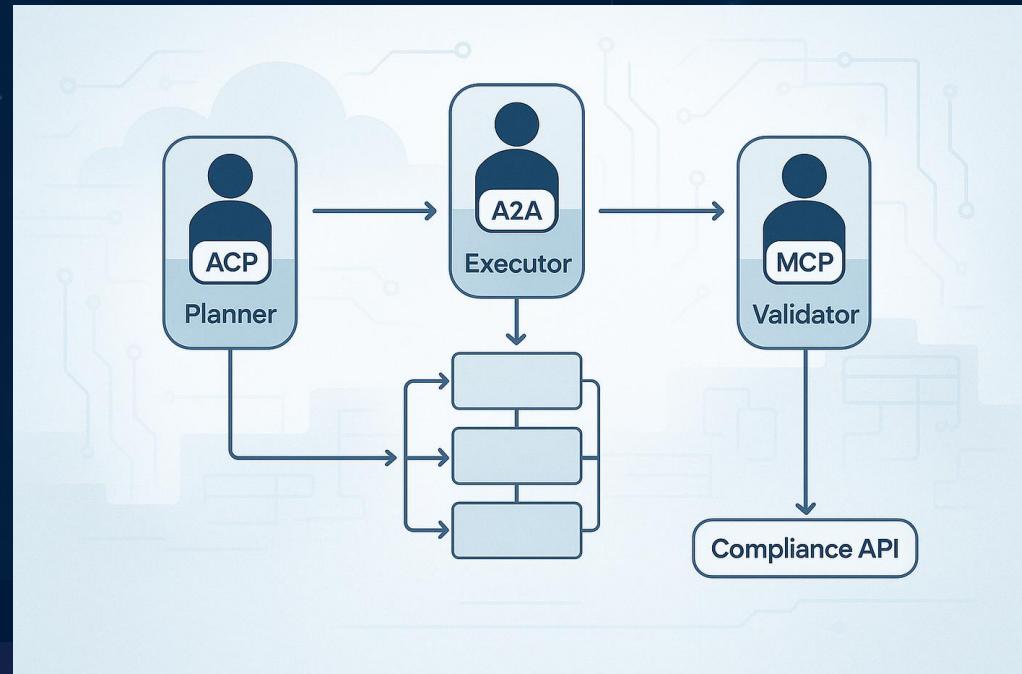
Protocols for task delegation, communication, conflict resolution, and resource sharing enable effective agent collaboration.

Learning and Consensus

Multi-agent reinforcement learning and consensus algorithms enhance coordination efficiency in dynamic environments.

When Multi-Agent Makes Sense

- ✓ Distinct reasoning roles
- ✓ Parallelizable work
- ✓ Long-running workflows
- ✓ Validation requirements



Hands-on Exercise: Designing Multi-Agent Workflows with No-Code and Low-Code Tools

Quick Pulse (Build – Test - Deploy)

QPulse is a simple

“ask → collect → summarize”

system that lets a facilitator post a question,
collect anonymous answers from a Telegram
chat, and generate a results summary
(including a winner + insights).

[Google Drive Link](#)

Quick Pulse

Where it's useful

- Team decisions: pick a restaurant, movie, activity, or hobby themes
- All-hands / retrospectives: quick anonymous feedback after a meeting
- Exit polls: immediate pulse on decisions or announcements
- Extended Family Vacation: Which destination to visit?
- Workshops: demonstrate multi-agent orchestration using real tools

Logistics & Core Components

- We will use N8N platform
 - Create an account in [n8n](#) for this project.
 - N8n has a 14-day trial account. That should be good enough.
- Models
 - N8N gives you trial openAI credits. That should be good enough.
 - Alternatively you can create a [OpenAI](#) account and fund it (\$5 should be good enough).
 - Alternate Model option: You can create account in [Google AI Studio](#) (for Gemini model)
 - Get API Keys for your models.
- Tools
 - We need a [Telegram](#) account
 - You can download the app on your mobile phone.

Lets Build you AI Agents and Workflows.

How it works (3 Workflows)

- 1) **AddQuestion** – A form posts a new question into QPulse_Questions Table
- 2) **UserResponse** – A Telegram ‘User Agent’ shares the current question and stores responses into QPulse_Answers Table
- 3) **GetResults** – A Multi-agent workflow (GateKeeper – Retriever – Summarizer) that fetches the latest open question, aggregates answers, and produces a summarized report.

Implementation Plan



Career Strategies & Growth in the Agentic AI Era

Long-Term Opportunities in AI

Infra AI (Enablement)

- Data Center
- Compute
- Network
- Model hosting
- Observability

Build AI (Intelligence)

- Models
- Training
- Evaluation
- Data Science
- Reasoning

Applied AI (Value Creation)

- FinTech
- HealthCare
- Legal
- Supply Chain
- Energy, Government

Building Credibility in Agentic AI



Technical Proficiency

Mastering AI/ML fundamentals, multi-agent systems, software architecture, and data engineering is essential for credibility.

Tool-driven Workflows

Problem-solving, cross-disciplinary collaboration, and continuous learning enhance effectiveness in agentic AI roles.

Measurable Automation

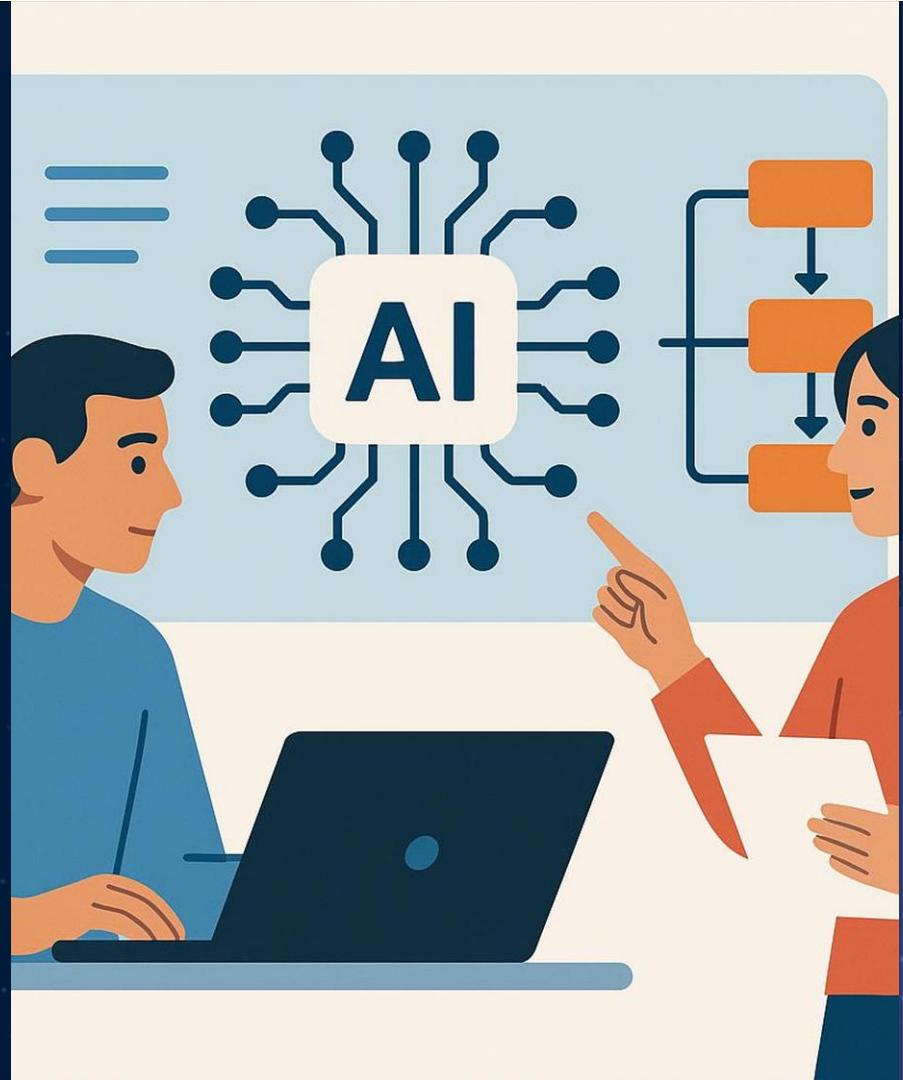
Playing around, trying things and contributing to open-source projects validate expertise and build professional credibility.

Path Forward for Software Professionals

- Shift from Writing Code to Designing Workflows
- Build Systems, Not Prompts
- Develop Hybrid Expertise (Software + AI + Domain)
- Failure handling

Mindset Change:

The future role isn't "AI replacing developers", it's developers becoming AI systems engineers who design how work gets done.



Conclusion: Embracing the Future with Agentic AI

- ❖ Agentic AI is an architecture shift
- ❖ Multi-agent mirrors distributed systems
- ❖ Most value is in orchestration, not models
- ❖ This is a software evolution moment
- ❖ Builders who adapt early will define this space

Thank
you