The AI Workforce Revolution: Leveraging Claude and Agentic Systems for Business

Executive Summary

The current technological landscape is shifting from fragmented AI chat sessions to integrated, autonomous **AI Workforces**. This transformation moves beyond using artificial intelligence as a simple "brain" for answering questions toward utilizing it as "hands" for executing complex business functions. Key industry experts emphasize that by 2028, approximately 15% of all work decisions will be made by AI agents.

This document synthesizes several advanced frameworks for building these systems, primarily focusing on **Claude Code**, **Claude Skills**, **Claude Plugins**, and platforms like **Relevance** and **OpenClaw**. The core thesis is that businesses can now deploy digital employees—specialized agents that collaborate in parallel—to handle entire departments such as marketing, sales, and operations. This "Business Operating System" approach solves traditional AI limitations regarding context memory, siloed sessions, and manual orchestration.

I. The Architecture of AI Workforces

A workforce is not a single chatbot but a coordinated team of specialized digital employees. Effective AI workforces are built on three fundamental pillars:

- **Specialization:** Each agent has one clear job (e.g., a researcher, a writer, a designer).
- **Collaboration:** Agents work together by passing tasks and information (e.g., a researcher passes a plan to a designer).
- **Coordination:** A central system or "orchestrator agent" manages the workflow to ensure everything runs smoothly without human intervention.

Comparative Framework: Skills vs. Plugins vs. Workforces

Feature	Claude Skill	Claude Plugin	AI Workforce
Scope	Single Task (e.g., Invoice generation)	Job Role (e.g., Marketer)	Entire Department (e.g., Marketing Team)
Analogy	A recipe card for one dish	An experienced chef	A fully staffed restaurant

Feature	Claude Skill	Claude Plugin	AI Workforce
Components	Custom instructions for one outcome	Bundled skills, tools, and connectors	Multiple agents, triggers, and knowledge bases
Persistence	Available in any chat	Persistent job function	End-to-end automated process

II. Key Technical Mechanisms

1. Memory and Context Management

Standard AI models often suffer from "context loss" or hallucinations as conversations grow. Advanced systems utilize specific file structures to maintain a "Business Brain":

- `claude.md` / `agent.md` : These serve as permanent instruction files or "system prompts" for specific folders. They allow the AI to "remember" the project context every time a session starts.
- **Hierarchical Settings:** Systems like Claude Code use a hierarchy of `settings.json` files (Local > Project > Global) to manage permissions, model selection, and tool access.
- **Progressive Disclosure:** A mechanism where Claude only reads parts of a skill or document as needed, optimizing the context window for better performance.

2. Specialized Agents and Sub-Agents

Instead of asking one AI to do everything, tasks are delegated to **Sub-agents**.

- **Isolated Context:** Each sub-agent often has its own context window (up to 200,000 tokens), preventing the "pollution" of the main session's memory.
- **Parallel Execution:** Claude can invoke multiple agents simultaneously. For example, a content team can have 14 sub-agents working on different social media platforms at once.

3. External Connectivity: API vs. MCP

To move beyond text generation, agents must connect to the real world.

- **APIs (Application Programming Interfaces):** Custom scripts allow agents to write to Google Docs, update CRMs like HubSpot, or post to social media.
- **MCP (Model Context Protocol):** A new, standardized "power adapter" for AI. It allows agents to immediately understand and use the tools available in an external app (e.g., the Playwright MCP for browser automation).

III. Advanced Implementation Platforms

Relevance AI

A visual "canvas" for building workforces. It allows for:

- **Triggers:** Starting a workforce via a Slack message or a scheduled time.
- **Inventing Tools:** Using natural language to generate Python code for new tools (e.g., a tool that turns transcripts into LinkedIn posts).
- **Knowledge Bases:** Syncing agents with Google Drive, Notion, or CSV files to provide specialized data.

OpenClaw (OpenClaw/OpenClow)

An open-source framework designed to run AI employees 24/7 on a server, often controlled via Telegram or Discord.

- **Cron Jobs:** "Alarm clocks" for AI, triggering tasks like a daily AI news digest at 8:30 AM.
- **Heartbeats:** Periodic check-ins (e.g., every 30 minutes) where the agent scans inboxes or calendars for new tasks.
- **Model Agnostic:** Can use various models, including Minimax (highly endorsed for its agentic capabilities at lower costs).

IV. Important Quotes with Context

> **"This is not about coding. It's about business leverage."** *Context: Daron Vener explaining that Claude Code, while built for developers, is the ultimate tool for non-technical business owners to run operations.*

> **"A skill is a recipe card... a plugin is an experienced chef who knows the whole recipe book."** *Context: Eliot Prince illustrating the difference between single-task instructions and multi-functional job roles within the Claude ecosystem.*

> **"The trick to creating effective skills is to make your AI think like an expert, not just follow steps."** *Context: Greg Isenberg quoting "Boring Marketer" on how to iterate on skills to ensure high-quality, professional-grade output rather than generic responses.*

> **"We are witnessing the birth of a trillion-dollar market... any industry will start to feel this shift."** *Context: Liam Otley discussing the rapid adoption of AI workforces and the opportunity for individuals to build and sell these systems to traditional businesses.*

V. Actionable Insights

1. **Start with "Claude Skills" for Operations:** Identify the top 10 repetitive tasks in the business (e.g., copywriting review, invoice generation, SOP writing). Create a specific Claude Skill for each to ensure consistency.

2. **Move from Chat Sessions to "Projects":** Stop using blank chat windows. Organize work into folders with a `claude.md` file to provide the AI with persistent memory of business goals and style guidelines.

3. **Implement an "AI Audit" for Clients:** For those looking to monetize this skill, offer a "Diagnose, Design, Deliver" model. Charge for a strategic audit that maps out exactly where AI workforces can replace manual workflows.

4. **Adopt "Plan Mode":** When using agentic tools, shift to "Architect" or "Plan Mode" first. This allows the user to review the AI's proposed multi-step strategy before it executes, ensuring safety and control.

5. **Leverage Cron Jobs for Routine Intelligence:** Use tools like OpenClaw to set up automated "research employees" that scan industry news or monitor company data while the team sleeps, delivering a summary every morning.

6. **Use Official MCPs for Stability:** While the Model Context Protocol is promising, experts suggest sticking to official, community-supported MCPs (like those from Microsoft or Anthropic) for reliability in a business environment.

Want to explore this topic further?

Book a free discovery call to discuss how ManaTech can help
your business implement these ideas.

Book a Discovery Call