Syntheticusers

From Prompts to Agents

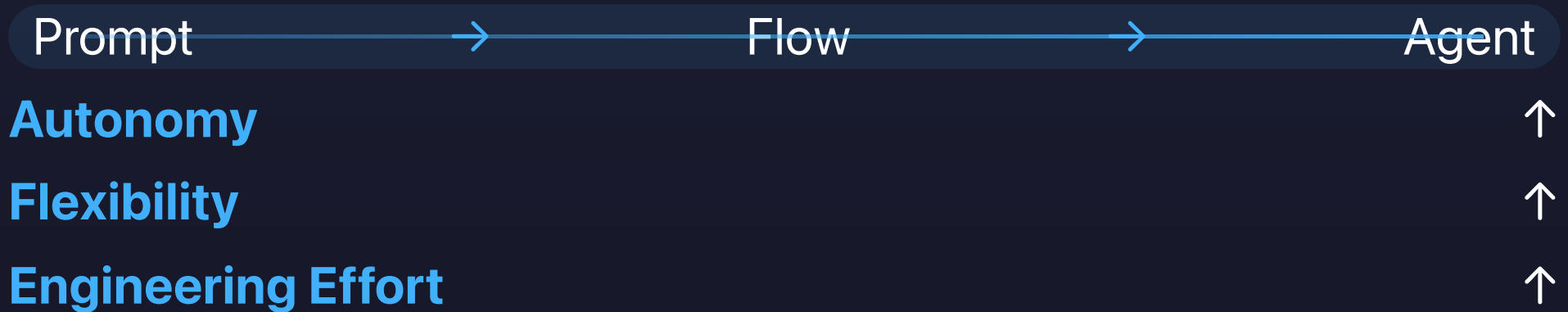
Building with AI at Different Levels of Autonomy

"We're shifting from AI as a tool to AI as a collaborator."

From Prompts to Agents: How AI is Learning to Run the Show

- **Synthetic Users** in action — showing:
  - Prompt-based testing of UX
  - Flow-driven user simulations
  - Agent-based behavior modeling
- Agenda: From prompts → flows → agents → future of autonomous organizations

The Evolution of AI Interactions



"We've gone from crafting smart outputs to designing AI behavior."

Prompt Engineering

- One-shot, stateless, human-in-the-loop
- Strengths:
  - Fast prototyping
  - Minimal code
  - Useful for content, transformation, Q&A
- Limitations:
  - No reasoning, memory, or decision-making
  - Breaks down with complexity or interactivity

Flow Engineering

- Multi-step, logic-driven workflows
- Structure = predictability and modularity
- Use Cases:
  - Structured decision trees
  - Templated document generation
  - LLM + rule-based automations
- Limitations:
  - Rigid and labor-intensive to maintain
  - Difficult to adapt to novel cases

Agentic AI – What Changes

- Agents operate in loops: **plan → act → observe → reflect**
- Key shift: from "generate text" to "decide what to do next"
- Benefits:
  - Handle ambiguity
  - Invoke tools and external systems
  - React to new info in real time

Axes of Relevance for AI Agents

> *"Not all agents are created equal. Here's how to think about their design space."*

1. **Autonomy** – from rule-following to self-directed
2. **Memory** – stateless vs task-level vs persistent agents
3. **Tool Use** – can the agent take action or just generate?
4. **Planning** – does it create plans or follow fixed ones?
5. **Human Involvement** – in-the-loop vs on-the-loop vs autonomous
6. **Specialization** – one skill or generalist?
7. **Coordination** – solo or part of a multi-agent system?

What Can Agents Actually Do?

- Core actions:
  - Query APIs
  - Execute code
  - Search the web
  - Access databases
  - Read/write files
  - Communicate with other agents
- Tools = agency
- Agent = planner + orchestrator

Agent Roles in a Modern Company

- **Support Agent:** Troubleshoots based on customer input, escalates edge cases
- **Ops Agent:** Syncs tools (CRM → Slack → dashboards), triggers alerts
- **Research Agent:** Collects data, summarizes insights, drafts reports
- **CRM Agent:** Follows up with leads, writes messages, updates records
- **UX Agent:** Tests product flows (as in **Synthetic Users**)
- Think of them as teammates with APIs instead of hands

Advantages of Agents

- **Flexibility** – adapt to new goals and environments
- **Scalability** – multiply productivity with minimal marginal cost
- **Emergent Behavior** – agents can surprise you with creative steps
- **Reusability** – same agent logic, many use cases
- **System Thinking** – unlock cross-functional workflows

The Future — Agents Running Companies?

- Can agents coordinate across departments?
- Thought experiment: A company of agents
  - Sales → CRM → Billing → Reporting
  - All agents, minimal humans
- Real examples emerging: startups using LLM agents for internal operations
- Analogy: What AWS did for infra, agents could do for ops

Challenges & Limits

- **Trust & Safety:** Hallucinations, bias, unpredictability
- **Observability:** Hard to debug "why" an agent made a choice
- **Governance:** Who's responsible when things go wrong?
- **Collaboration:** How do humans + agents work as a team?
- **Evaluation:** How do we measure success?
  - Metrics: Task success rate, autonomy %, recovery from failure, resource usage
  - Evaluation = continuous monitoring + structured feedback
- Key: Keep humans in the loop for oversight and accountability

Strategic Guidance

- Crawl → Walk → Run strategy:
  - Start with prompts
  - Build flows
  - Scale to agents when tasks require autonomy
- Pair agents with humans at first (copilot model)
- Monitor, tune, and slowly expand their decision space

Final Thoughts

- Agents = productized intelligence
- Not magic — they're structured loops with memory + tools
- Design them like you'd build a team:
  - Give them roles, permissions, and reviews
  - Let them specialize, then collaborate
- Start small. Iterate. Let them grow into it.

Q&A / Contact

- Q&A

  Contact: hugo@syntheticusers.com | LinkedIn
- Ask about **Synthetic Users** and how we use agents internally

Syntheticusers