Search as Code: Why This May Be a Bigger Shift Than It First Appears

Date: June 2026
Source: Perplexity Research – "Rethinking Search as Code Generation"

Executive Summary

Perplexity's "Search as Code" concept is not really about making search faster. The more important idea is that AI systems are evolving from **tool users** into **workflow generators**.

Traditional AI agents operate by making a series of tool calls:

Question → Search → Wait → Search → Wait → Analyze → Answer

Perplexity proposes a different approach. Instead of repeatedly calling tools, the AI generates a small program that orchestrates the retrieval process itself.

The result is fewer round trips, cleaner context, greater parallelization, and a more natural way for AI systems to solve complex information problems.

Traditional Agent Pattern

Most current AI agents work like a person manually completing tasks one at a time.

Example:

1. List Florida counties
2. Search NOAA flood data for Miami-Dade
3. Search NOAA flood data for Lee
4. Search NOAA flood data for Collier
5. Search NOAA flood data for Pinellas
6. Retrieve FEMA NRI scores
7. Join datasets
8. Rank results

Each action requires a separate tool call.

This creates:

- Additional latency
- More context consumption

- More opportunities for failure
- More complexity managing state

Search as Code Pattern

Instead of making eight separate requests, the AI generates a workflow:

```
counties = get_counties("FL")

flood = parallel_search(
    counties,
    "NOAA flood"
)

risk = parallel_search(
    counties,
    "FEMA NRI"
)

joined = join(
    flood,
    risk,
    key="FIPS"
)

answer = rank(
    joined,
    by="flood_score"
)
```

The AI becomes the workflow architect.

The system executes the workflow.

The result arrives in a single orchestration cycle.

Why This Matters

The speed increase is interesting.

The architectural change is more important.

The AI is no longer just answering questions.

It is designing processes.

This moves AI systems closer to how software engineers think:

- Gather data
- Transform data
- Join datasets
- Analyze results
- Produce outputs

Rather than repeatedly asking tools for information.

Connection to GIS and Red Cross Work

Many GIS and disaster response workflows already follow this model.

Examples:

Disaster Assessment

Damage Assessments → Census Data → SVI → FEMA NRI → CRCI → Shelter Data → County Aggregation → Operational Report

Situational Awareness

Weather Feeds → County Boundaries → Population Data → Vulnerability Layers → Impact Assessment → Decision Brief

Volunteer Operations

Volunteer Database → Chapter Assignments → Availability → Skill Matching → Deployment Recommendations

These are all workflow problems rather than search problems.

Search as Code suggests future AI systems will dynamically generate these workflows.

Relationship to My Current Development Work

Many of my recent projects already reflect this architecture:

- Python pipelines
- ArcGIS data enrichment
- Geocoding systems
- Hosted feature layer updates
- Automated weather reporting
- Data joins using FIPS codes
- FastAPI applications
- GitHub Actions workflows
- Vercel-hosted tools

The difference is that today I design the workflow manually.

Future agent systems may generate significant portions of these workflows automatically.

Historical Perspective

Technology progression often follows this pattern:

Manual Tasks → Scripts

Scripts → APIs

APIs → AI-Orchestrated Workflows

Search as Code appears to be another step in that evolution.

The destination is likely not "better chatbots."

The destination is AI systems that can design, execute, monitor, and adapt complex operational workflows.

Key Takeaway

The most important idea from Search as Code is not that it makes search faster.

The important idea is that AI is increasingly becoming an orchestration layer capable of generating workflows instead of simply calling tools.

For GIS, disaster operations, logistics, planning, and decision support systems, this may prove to be one of the most significant architectural shifts of the current AI era.

Personal Observation

The future of AI may be less about conversations and more about orchestration.

The winning systems may not be those that answer questions best.

They may be the systems that can design and execute the most effective workflows.