MSIS 549 HW2: LinkedIn Thought Leadership Agent

Tutorial Write-Up

Author: Daa Al-Ashtawy **Course**: MSIS 549 — AI and GenAI for Business Applications **Instructor**: Prof. Leonard Boussioux **Date**: February 2026 **Path**: A (Skills Pack) | **Bonus**: MCP Tool (LiGo LinkedIn Runner)

System Artifact: GitHub Repository

Table of Contents

---

1. Problem Statement

The Pain Point

Senior leaders (Managers, Directors, VPs) have deep domain expertise but consistently struggle to maintain a visible presence on LinkedIn. The typical failure pattern:

1. **Week 1**: Leader writes a thoughtful post, gets 50+ reactions, feels motivated.
2. **Week 2-3**: Calendar fills up. No post. Momentum lost.
3. **Week 4+**: The "I should post more on LinkedIn" guilt grows, but the activation energy to plan, draft, and refine a post from scratch is too high.

The core problem isn't writing ability — it's the **overhead of planning a cohesive series** that builds on itself week over week. A single post is manageable; a 6-week narrative arc that positions the leader as a thought leader requires sustained effort that competes with their day job.

Why an Agentic Workflow?

A single-prompt LLM request ("write me 6 LinkedIn posts about SQL") produces generic, disconnected content that sounds like AI and doesn't reflect the leader's unique voice or experience. What's needed is a **multi-stage pipeline** that:

- **Interviews** the leader to extract their authentic voice and experience (5 minutes of their time)
- **Plans** a cohesive 6-week narrative arc (not 6 random posts)
- **Drafts** with anti-AI-ism rules built in
- **Refines** to match the leader's specific tone (provocative, educational, etc.)
- **Optimizes** for LinkedIn's algorithm and formatting
- **Benchmarks** quality against defined criteria
- **Archives** for reuse and learning

Each of these stages requires different expertise — which maps naturally to specialized skills/agents working in sequence.

Tech Stack

- **Skills**: 9 Markdown files (`.md`) defining behavior, constraints, and prompts
- **Orchestration**: Manus AI (primary), Claude Code (secondary testing)
- **MCP**: LiGo LinkedIn Runner for optional direct publishing
- **Output**: Markdown archive files, LinkedIn-ready post text
- **Repository**: GitHub

---

2. System Design

2.1 Architecture Overview

The system is a **sequential pipeline** with 9 specialized skills, coordinated by a Master Orchestrator (Skill 0). Two mandatory Human-in-the-Loop (HITL) checkpoints ensure quality before finalization.

> ***See***: `diagrams/architecture_infographic.html` *for the full visual architecture diagram.*

2.2 All 9 Skills

#	Skill Name	File	Purpose	Key Innovation
0	**Master Orchestrator**	`skill_0_master_orchestrator.md`	Sequences all skills, maintains state, enforces HITL gates	Error recovery + state management
1	**Intent Discovery**	`skill_1_intent_discovery.md`	5-question strategic interview	Structured extraction of voice/audience/anecdote
2	**Content Strategist**	`skill_2_content_strategist.md`	Creates 6-week narrative arc roadmap	Proven Hook→Framework→Story→Tactics→Vi structure
3	**Draft Architect**	`skill_3_draft_architect.md`	Generates initial 150-300 word drafts	Built-in anti-AI-ism rules
4	**Voice & Tone Refiner**	`skill_4_voice_tone_refiner.md`	Ghostwriter: matches leader's authentic voice	AI pattern detection + replacement ta
5	**Engagement Optimizer**	`skill_5_engagement_optimizer.md`	LinkedIn formatting, hooks,	Platform-specific optimization

			hashtags + **HITL 5a**	
6	**Quality Reviewer**	`skill_6_quality_reviewer.md`	5-metric benchmark + **HITL 6a**	Structured scoring with failure flaggin
7	**Archive Manager**	`skill_7_archive_manager.md`	Packages complete session to Markdown	Full reproducibility of every run
8	**Poster & Reviewer**	`skill_8_poster_reviewer.md`	Publishes via LiGo MCP (LinkedIn API)	**MCP bonus**: direct LinkedIn integratic

2.3 Pipeline Flow

```
User Input (Topic)
    │
    ▼
┌─────────────────┐
│ Skill 0: Master  │ ← Maintains state across all phases
│   Orchestrator   │
└────────┬────────┘
         │
    ▼ Phase 1: Discovery
┌─────────────────┐
│ Skill 1: Intent  │ ← 5-question interview
│   Discovery      │
└────────┬────────┘
         │ Strategic Intent Document
         ▼ Phase 2: Strategy
┌─────────────────┐
│ Skill 2: Content │ ← 6-week roadmap
│   Strategist     │
└────────┬────────┘
         │ 6-Week Roadmap
         ▼ Phase 3: Creation (3 sub-stages)
┌─────────────────┐
│ Skill 3: Draft   │ ← Initial drafts
│   Architect      │
└────────┬────────┘
         │
┌─────────────────┐
│ Skill 4: Voice   │ ← Voice refinement
│   Refiner        │
└────────┬────────┘
         │
┌─────────────────┐
```

```
│ Skill 5: Engage  │ ← LinkedIn optimization
│   Optimizer      │
└────────┬─────────┘
         │ 6 Optimized Posts
         ▼ Phase 4: HITL Checkpoint 5a
┌──────────────────┐
│ ✋ USER REVIEW    │ ← Approve / Revise / Reject each post
│   (HITL 5a)      │ ← Loop back to Skills 3-5 if needed
└────────┬─────────┘
         │ Approved Posts
         ▼ Phase 5: Benchmark
┌──────────────────┐
│ Skill 6: Quality │ ← Score 5 metrics, present scorecard
│   Reviewer       │
└────────┬─────────┘
         │
┌──────────────────┐
│ ✋ USER REVIEW    │ ← Approve benchmark results (HITL 6a)
│   (HITL 6a)      │
└────────┬─────────┘
         │ Approved Series + Scores
         ▼ Phase 6: Finalization
┌──────────────────┐      ┌──────────────────┐
│ Skill 7: Archive │      │ Skill 8: LiGo    │
│   Manager        │      │   MCP Publisher  │ (optional)
└──────────────────┘      └──────────────────┘
```

2.4 Key Design Decisions

1. **Sequential pipeline over parallel agents**: LinkedIn posts must form a coherent narrative arc. Parallel generation would produce disconnected posts. The sequential flow ensures each stage builds on the previous one's output.

2. **Two HITL checkpoints, not one**: Checkpoint 5a catches content/voice issues early (before benchmark effort). Checkpoint 6a gives the user confidence in quality metrics before archiving. This prevents wasted computation on posts the user would reject.

3. **Separate Voice Refiner (Skill 4) from Draft Architect (Skill 3)**: The initial draft focuses on structure and substance. Voice refinement is a separate concern — this separation of concerns makes it easier to iterate on voice without re-drafting content.

4. **MCP as optional Skill 8**: Not all users will want automated posting. Making it optional (and the last step) means the core pipeline works without MCP configuration.

3. Building Process

3.1 Tools & Timeline

Step	Activity	Tool	Time
1	Research LinkedIn best practices, skill format	Web research, Claude documentation	45 min

2	Design the pipeline architecture	Mermaid diagrams, whiteboard sketching	1 hr
3	Write Skills 0-7 (first draft)	Manus AI + manual editing	2 hrs
4	Test pipeline end-to-end on SQL topic	Manus AI orchestration	1.5 hrs
5	Iterate on Skills 3-4 (voice quality issues)	Manual prompt refinement	1 hr
6	Add Skill 8 (LiGo MCP integration)	LiGo MCP documentation	30 min
7	Run second test case (Data Cleanliness)	Manus AI	45 min
8	Design benchmark, run scoring	Manual evaluation	1 hr
Total			**~8.5 hrs**

3.2 Key Bottlenecks

Bottleneck 1: Narrative Cohesion The hardest challenge was making 6 posts feel like chapters of a book rather than 6 random articles. The first version of Skill 2 (Content Strategist) produced weekly themes that were loosely related but didn't build on each other. The fix was defining the explicit narrative arc: Hook → Framework → Story → Tactics → Vision → Call. This structure forces each week to serve a specific purpose in the overall narrative.

Bottleneck 2: AI Voice Detection Early drafts from Skill 3 consistently started with "In today's fast-paced world" or "Let's dive in." Simply telling the LLM to "avoid AI-isms" wasn't enough. The fix was creating an explicit detection-and-replacement table in Skill 4 (see Section 4.1).

Bottleneck 3: HITL Feedback Integration When the user requested changes during HITL 5a, it wasn't clear which skill to re-invoke. A voice issue needed Skill 4; a structure issue needed Skill 3. The fix was adding routing logic to the Orchestrator: voice/tone changes → Skill 4, content/structure → Skill 3, formatting → Skill 5.

3.3 Decisions Along the Way

- **Why not n8n?** I considered Path B (n8n) but chose Path A (Skills Pack) because: (a) the workflow is interactive and requires human judgment at two stages, which is harder to orchestrate in a visual workflow builder; (b) skills are more portable — I can use them in Claude Code, Manus, or any LLM assistant.
- **Why 9 skills instead of 3-4?** Each skill has a single, clear responsibility. This made debugging much easier — when the voice sounded wrong, I knew to look at Skill 4, not dig through a monolithic prompt.
- **Why Manus AI?** Manus provided a clean environment for orchestrating multi-skill pipelines with state management. I also tested the skills in Claude Code to verify portability.

---

4. Prompt Iteration & Critique

4.1 Iteration 1: Voice & Tone Refiner (Skill 4)

Version 1 (Initial):

```
Rewrite the drafts to sound professional.
```

Problem: The LLM interpreted "professional" as "corporate press release." Posts became stiff and impersonal.

Version 2 (Iteration):

```
Rewrite the drafts to sound like a senior leader (Manager/Director). Remove AI-isms
like 'In today's fast-paced world' and 'Let's dive in.' Use strategic phrasing like
'operational efficiency' and 'competitive advantage.'
```

Problem: Better, but "strategic phrasing" became a crutch. Every other sentence used "operational efficiency" or "competitive advantage."

Version 3 (Final — shipped):

```
Act as a ghostwriter for a senior leader. Apply these specific rules:

REMOVE (AI patterns):
| Pattern | Replace With |
| "In today's fast-paced world" | [Delete — start with specific claim] |
| "Let's dive in" | [Delete — just start content] |
| "It's important to note" | [Delete — if important, reader will see it] |
| "This is a game-changer" | [Replace with specific impact: "cut query time by 60%"] |

ADD (professional weight):
- Use "Strategic alignment" not "matching goals"
- Use "Time-to-market" not "getting things done quickly"
- Use first person: "I" and "we" — leaders speak from experience

MATCH TONE:
- If Provocative: bold claims, rhetorical questions
- If Educational: numbered steps, "here's how"
- If Empathetic: "I've been there" language
- If Data-driven: lead with statistics
```

Result: The explicit detection table was the breakthrough. Instead of vaguely asking the LLM to "avoid AI-isms," the table gives concrete patterns to find and replace. Voice consistency scores jumped from 3.0 to 4.7.

4.2 Iteration 2: Content Strategist (Skill 2)

Version 1: "Create a 6-week plan with different angles for each week."

Problem: Weeks were related to the topic but felt like 6 standalone posts with no narrative progression.

Version 2 (Final): Defined the explicit arc:

```
Week 1: The Hook (Pain Point) — surface a problem the audience recognizes
Week 2: The Framework — introduce a mental model
Week 3: The Story — share the personal anecdote
Week 4: The Tactics — concrete, actionable advice
Week 5: The Vision — connect to industry future
Week 6: The Call — synthesize + drive the CTA
```

Result: Narrative Cohesion scores jumped from 2.5 to 4.8. The arc creates natural "coming next week" teasers because each week logically leads to the next.

4.3 Prompt Quality Critique

Strengths:

- Constraints are specific and measurable (e.g., "150-300 words," "3-5 hashtags")
- The detection table in Skill 4 is concrete — no ambiguity about what to remove
- HITL checkpoints are clearly defined with explicit user prompts

Weaknesses:

- Skills rely heavily on the quality of Skill 1's interview. If a user gives vague answers, all downstream skills produce weaker content (see edge case in benchmark).
- The voice profile is hard-coded to "Manager/Director." A technical audience (e.g., database kernel engineers) gets inappropriate simplification.
- No fallback for when the LLM's training data doesn't cover a niche topic deeply enough.

---

5. Real Usage & Iteration

5.1 Run 1: SQL Query Performance (Primary Test Case)

Input: Topic = "Optimizing SQL Query Performance" for Data Engineers and Business Stakeholders. Educational + Provocative tone. Personal anecdote about a business team frustrated with data lag.

Process: Full pipeline execution through Skills 0-7.

Output: 6-week series saved to `outputs/Q1_2026_SQL_Performance_Series.md`. See full posts in the output file.

What Worked:

- The 6-week narrative arc felt cohesive — each post naturally led to the next
- Personal anecdote in Week 3 was the strongest post (scored 4.8/5.0)
- Hooks were strong: "Is your data platform actually 'good enough'?" performed well

What Didn't Work:

- Week 5 (AI-Ready Infrastructure) was the weakest — too vague, not enough specific examples
- Some posts still had subtle AI patterns ("In the era of AI" in Week 1)
- Visual recommendations were generic (stock photo descriptions rather than specific diagram specs)

Changes After Run 1:

- Strengthened the anti-AI-ism list in Skill 4 to catch "In the era of..." pattern
- Added "system diagram" suggestions alongside visual recommendations in Skill 5
- Revised Week 5 draft to include specific architectural examples

5.2 Run 2: Data Cleanliness (Second Test Case)

Input: Topic = "Enhancing Data Cleanliness" for Data Analysts and Product Managers. Educational + Empathetic tone. Anecdote about a product launch delayed 2 weeks due to dirty customer data.

Process: Full pipeline with the improvements from Run 1.

Output: 6-week series (not included in full — see benchmark appendix for titles and scores).

What Worked:

- The empathetic tone was noticeably different from Run 1's provocative tone — Skill 4 correctly adapted
- The product launch anecdote in Week 3 was compelling and specific
- Week 4 ("5 Data Quality Checks") was highly actionable

What Didn't Work:

- Week 5 was again the weakest — the "vision" week tends to become generic
- The system didn't challenge the user when the initial CTA was vague ("improve data quality")
- Run 2 took less time (confirming the pipeline is reusable) but still required HITL revisions on 2/6 posts

Changes After Run 2:

- Added a prompt in Skill 1 to push back on vague CTAs: "Can you make that more specific? What's the ONE thing you want readers to do this week?"
- Noted that Week 5 (Vision) consistently underperforms — potential improvement is adding a "trends research" sub-step

---

6. Benchmarking Methodology & Results

6.1 Methodology

Approach: Human Rubric Scoring (Method 1) + Baseline Comparison (Method 3)

Baseline: Single-prompt GPT-4 request: "*Write 6 LinkedIn posts about SQL Query Performance for Data Engineers and Business Stakeholders.*"

5 Metrics (each scored 1-5):

1. Actionability — concrete next steps for the reader
2. Voice Consistency — sounds like the same leader across all 6 posts
3. Strategic Depth — genuine expertise, not surface-level
4. Narrative Cohesion — posts build on each other
5. LinkedIn Optimization — hooks, formatting, hashtags

Test Cases: 4 total — 2 standard, 1 edge case (highly technical topic), 1 ambiguous case (vague input).

> ***Full scoring tables, baseline outputs, and failure analysis***: *See* `benchmark/BENCHMARK_APPENDIX.md`

6.2 Summary Results

Test Case	Agentic Score	Baseline Score	Delta
SQL Performance	**4.5**	2.0	+2.5
Data Cleanliness	**4.2**	—	—
Edge: Technical Topic	**3.4**	—	—
Ambiguous: Vague Input	**3.6**	—	—

Biggest win: Narrative Cohesion — the baseline scored 1.0 (posts are disconnected), the agentic system scored 4.8 (posts form a story). This is the single largest improvement and validates the core design decision of using a 6-week roadmap skill.

Worst failure: Edge case, Week 5 — Voice Consistency scored 2.0 because Skill 4 replaced precise technical terminology with executive language, which was inappropriate for database kernel engineers.

6.3 What the Benchmark Revealed

1. **The interview is everything**. Skill 1 (Intent Discovery) determines the ceiling for the entire series. Rich, specific inputs → excellent output. Vague inputs → mediocre output.
2. **The "Vision" week (Week 5) is consistently the weakest**. It tends toward generic futurism. Potential fix: add a research step that pulls recent industry articles/trends.
3. **Voice refinement works well for general audiences, poorly for niche technical audiences**. The "Manager/Director" voice profile is effective for most use cases but needs audience-specific variants.
4. **Two HITL checkpoints caught issues that automated scoring missed**. The user caught a tone issue in Run 1 HITL 5a that the benchmark metrics in Skill 6 wouldn't have flagged (the post was technically well-written but didn't "feel" like the leader's voice).

---

7. Reflection

What Worked Well

- **The pipeline structure is genuinely reusable.** After building the skills once, Run 2 took significantly less time than Run 1. The interview → roadmap → draft → refine → review flow works for any topic.
- **HITL checkpoints are essential.** Full automation would produce content the leader wouldn't publish. The checkpoints respect the leader's judgment while saving them 90% of the effort.
- **Anti-AI-ism rules in Skill 4 made the biggest difference.** The detection table approach is more effective than vague instructions like "sound natural."

What Didn't Work

- **Week 5 (Vision) is a consistent weak spot.** The system doesn't have access to real-time industry data, so the "future" week relies on the LLM's training data, which may be stale.
- **Voice Refiner is one-size-fits-all.** It needs audience-specific profiles (executive, technical expert, practitioner).
- **No multi-format output.** The system produces text only. A future version could generate Canva-ready visuals or video scripts.

How Prompts Evolved

The biggest learning was that **specific, table-based constraints outperform vague instructions**. "Sound professional" is useless. A table that says "replace X with Y" is actionable for the LLM. This applies broadly to any skill-based system.

Would I Keep Using This?

Yes. I plan to use this system for my own LinkedIn presence in Q1 2026. The SQL Performance series (Run 1) is ready to publish. The system saves approximately 4-5 hours per 6-week series compared to writing from scratch, while producing content that's more strategically cohesive than what I'd write in scattered 30-minute sessions.

---

8. How to Replicate

Prerequisites

- An LLM assistant that supports Markdown skills (Claude Code, Manus AI, or similar)
- (Optional) LiGo MCP server configured for LinkedIn publishing

Step-by-Step

1. **Clone the repository**:

   ```
   git clone https://github.com/taashim-eng/linkedin-thought-leadership-agent.git
   ```

2. **Install the skills** in your LLM's skills directory:
   - Claude Code: Copy `skills/*.md` to `~/.claude/skills/linkedin-agent/`
   - Manus AI: Upload the skills via the platform interface

3. **Start a session** with this prompt:

   ```
   I want to create a 6-week LinkedIn thought leadership series on [YOUR TOPIC].
   Please use the LinkedIn Thought Leadership Orchestrator to guide me through the
   process.
   ```

4. **Complete the 5-question interview** (Skill 1 will ask you).

5. **Review the 6-week roadmap** (Skill 2 output).

6. **Wait for drafts** (Skills 3-5 run sequentially).

7. **Provide feedback at HITL 5a** — approve, revise, or reject each post.

8. **Review benchmark scores at HITL 6a** — approve or request changes.

9. **Find your final series** in the `archive/` directory.

MCP Setup (Optional — for auto-publishing)

Add to your LLM's MCP configuration:

```
{
  "mcpServers": {
    "ligo-linkedin": {
      "command": "npx",
      "args": ["-y", "@anthropic/ligo-mcp-server"],
      "env": {
        "LINKEDIN_CLIENT_ID": "<your-id>",
        "LINKEDIN_CLIENT_SECRET": "<your-secret>"
      }
    }
  }
}
```

Total word count: ~2,800 words | Estimated reading time: 12 minutes