ChatGPT

Systematic Literature Review Results

RQ1: *To what extent is implementing agentic AI workflows necessary for automated proposal generation?*
RQ1.1: *Which agentic AI design patterns are essential for realizing such workflows?*

The review (2023–2025) identified 20 top-ranked papers (all in English) across insurance, legal, software engineering, sales, and related enterprise domains. Each study provides architectural details or evaluations of multi-step **agentic workflows** (as opposed to single-shot LLM prompts) and was assessed with the Parsifal SLR quality checklist (cutoff score = 2.5/6.5). Below is a ranked table of the selected papers, including QA score and main contribution (i.e. key **agentic design pattern** or finding):

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
1	**Automated Design of Agentic Systems (ADAS)** (Hu et al., 2024)	6.5	Introduces a *meta-agent search* algorithm that **automatically invents new multi-agent workflow designs**, treating agent behaviors as code to be evolved and optimized [1]. Demonstrates that learned agentic workflows can outperform hand-crafted ones, underscoring the **necessity of automated agent design** for complex tasks.	arXiv: 2408.08435
2	**Routine: A Structural Planning Framework for LLM Agents in Enterprise** (Zhang et al., 2025)	6.5	Proposes a **plan-then-act** framework ("Routine") with a well-defined, *structured planning language* for multi-step tool use. The planner produces a formalized plan (Routine) passed to an executor agent, enabling reliable tool calls. In a real enterprise scenario, Routine vastly improved execution success (e.g. GPT-4 from 41% to 96% accuracy) [2]. This showcases a **pattern of explicit workflow planning** that greatly enhances stability in enterprise quote-generation processes.	arXiv: 2507.14447

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
3	**Agentic AI: A Comprehensive Survey of Architectures, Applications, and Future Directions** (Abou Ali et al., 2025)	5.5	Provides a **systematic review and dual-paradigm taxonomy** of agentic AI. Distinguishes single-agent vs. multi-agent approaches, noting that *single LLM agents can autonomously draft proposals, but teams of specialized agents collaborating (e.g. manager, researcher, writer, QA) produce more thorough results* [3]. Identifies common design patterns (prompt chaining, role specialization, tool use, memory) and observes that **agentic workflows are increasingly crucial** for complex, high-stakes domains (finance, legal, etc.) where single-shot LLMs fall short.	arXiv: 2510.25445
4	**MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework** (Hong et al., 2023)	4.0	Pioneers a multi-agent system that encodes **Standardized Operating Procedures (SOPs)** into prompts. Uses an *"assembly line"* of specialized agents (e.g. product manager, engineer, tester) to break down software project requests into sub-tasks [4]. This role-based collaboration pattern reduced logical errors by having agents verify each other's outputs, proving the **necessity of structured multi-agent workflows** (over a single chat agent) for complex tasks like software proposal generation.	arXiv: 2308.00352
5	**AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation** (Wu et al., 2024)	4.0	Introduces a general-purpose **conversation-driven multi-agent framework**. Agents converse in natural language or code, and developers can program flexible interaction patterns [5]. AutoGen demonstrates that multi-agent dialogues (LLM agents + tools + optional human inputs) solve tasks in math, coding, and ops planning better than single-agent prompts. The pattern of *"agents that talk to each other"* shows high adaptability, confirming agentic workflows' benefit for dynamic proposal generation in diverse domains.	arXiv: 2308.08155

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
6	**Nexus Architect: Adaptive Multi-Agent Reasoning via Automated Workflow Generation** (Barbosa et al., 2025)	4.0	Enhances complex reasoning through a **hierarchical agent organization**. A *"Supervisor"* agent dynamically spawns and coordinates specialist sub-agents (tool users, reasoners) in a workflow [6] [7]. Introduces an iterative feedback loop ("Iterative Prompt Refinement") where agents critique and refine solutions. This demonstrates a **divide-and-conquer pattern** (global manager + specialized workers) that is necessary when generating complex proposals requiring reasoning, tool use, and result verification beyond the capability of a single LLM.	arXiv: 2507.14393
7	**LightAgent: Production-Level Open-Source Agentic AI Framework** (Cai et al., 2025)	4.0	Develops a lightweight, **modular agent framework** emphasizing robustness and scalability. Integrates core components (memory, tool-use, and a built-in *Tree-of-Thought* reasoning module) in a minimal codebase [8]. Supports automated tool generation and fault-tolerant agent cooperation ("LightSwarm"). LightAgent demonstrates the *error-correction* pattern (agents detect/correct hallucinations) and shows that **enterprise-grade agentic systems** can be efficient, suggesting the practicality of agentic workflows in production (e.g. for sales or insurance proposals).	arXiv: 2509.09292
8	**Flow: A Modular Approach to Automated Agentic Workflow Generation** (Niu et al., 2025)	4.0	Proposes dynamically **modifiable multi-agent workflows**. "Flow" represents tasks as an Activity-on-Vertex graph and emphasizes *modularity* and *parallelism* [9]. During execution, the workflow can be **updated in real-time** (reallocating sub-tasks or altering agent roles if failures occur) [10]. This pattern of **dynamic workflow adjustment** (vs. a static chain) yielded dramatic efficiency gains in case studies (e.g. concurrent coding tasks). It underlines that for complex proposal generation (which may face changing requirements), an adaptive agentic workflow is more effective than a static plan.	arXiv: 2501.07834

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
9	**AFlow: Automating Agentic Workflow Generation** (Zhang et al., 2025)	4.0	Views an agent workflow as a *searchable program graph*. AFlow uses **Monte Carlo Tree Search (MCTS)** to automatically generate and optimize workflows (nodes are LLM calls/tools; edges are data flow) [11] . It iteratively refines candidate workflows based on execution feedback. Across benchmarks, AFlow's AI-designed workflows outperformed human-written ones, even enabling smaller LLMs to beat a GPT-4 baseline [12] . The key pattern is treating workflow construction as **search/optimization**, highlighting that automated search can find more effective proposal-writing strategies than manual design, validating the necessity of agentic optimization.	[arXiv: 2410.10762](https://arxiv.org/abs/2410.10762)
10	**A^2Flow: Automating Agentic Workflow Generation via Self-Adaptive Abstraction Operators** (Zhao et al., 2025)	4.0	Extends workflow automation by learning **abstract operators**. A^2Flow automatically derives reusable high-level actions through a three-stage process (case-based operator generation, clustering, and chain-of-thought–guided abstraction) [13] . These learned operators serve as building blocks for workflows, avoiding manual definition of sub-tasks. This introduces a **hierarchical abstraction pattern** – the agent can plan at a higher level and fill in details – which improves generalization and efficiency. It suggests that for proposal generation, giving an agent abstract actions (e.g. "gather client requirements") yields more scalable solutions than hard-coding every low-level step.	[arXiv: 2511.20693](https://arxiv.org/abs/2511.20693)

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
11	**MaAS: Multi-Agent Architecture Search via Agentic Supernet** (Guibin Zhang et al., 2025)	4.0	Introduces a **neural architecture search** approach for agent teams. MaAS trains an *"agentic supernet"* that encompasses a vast space of possible multi-agent system configurations [14]. For each new query/task, it then samples a tailored agent organization from this supernet, trading off performance vs. cost. This approach – **automatically configuring the agent society per task** – showed up to 11% performance boosts with 6–45% less cost. It highlights that the *optimal workflow structure can vary by proposal*, and having a mechanism to auto-tune the agentic workflow to the task at hand is highly beneficial.	[arXiv: 2502.04180](https://arxiv.org/abs/2502.04180)
12	**HuggingGPT: Solving AI Tasks with ChatGPT and its Friends (Hugging Face Models)** (Shen et al., 2023)	4.0	Demonstrates the **LLM-as-orchestrator** pattern. A central LLM (ChatGPT) serves as a *controller* that plans tasks and delegates them to specialist models (vision, speech, etc.) via HuggingFace APIs [15]. HuggingGPT translates a user request into sub-tasks, chooses appropriate expert models, and then integrates their outputs. This tool-using agent workflow significantly outperformed single models on multi-modal tasks. It provides evidence that **proposal generation can require diverse capabilities** (data extraction, calculation, visualization) and that an agentic approach (LLM "manager" + tool experts) is necessary to cover those skills.	[arXiv: 2303.17580](https://arxiv.org/abs/2303.17580)

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
13	**Reflexion: Language Agents with Verbal Reinforcement Learning** (Shinn et al., 2023)	4.0	Proposes a **self-reflection feedback loop** for a single LLM agent. The agent maintains an *episodic memory* of its past actions and outcomes, and after each trial it **verbalizes a critique (reflection)** of errors, storing insights in memory [16]. These reflections then guide the agent in the next attempt, dramatically improving performance on coding and decision-making tasks (e.g. surpassing GPT-4 on HumanEval coding with 91% accuracy [17]). This pattern – *learn from past mistakes via self-feedback* – is essential for proposal generation, where an agent can refine a draft proposal through iterative critique, something a single-pass LLM cannot do.	arXiv: 2303.11366
14	**DocAgent: An Agentic Framework for Multi-Modal Long-Context Document Understanding** (Sun et al., 2025)	4.0	Presents a **multi-agent reading system** that imitates how humans analyze long documents (e.g. contracts or RFPs). One agent first creates a structured outline of the document, and others query different sections via an interactive reading interface [18]. A *Reviewer agent* cross-checks answers using external sources, and a shared memory bank stores knowledge for reuse. This demonstrates **decomposition and parallelization** in comprehension – multiple specialized agents focus on different parts (text, tables, images) – which is crucial for proposal generation when input materials are lengthy and complex. The pattern ensures completeness and accuracy in understanding source documents before drafting a proposal.	ACL Anthology ID: 2025.EMNLP-main.893

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
15	**Agentic Workflow for Healthcare Simulation Scenario Design** (Costa et al., 2025)	4.0	Describes a domain-specific **agentic workflow** that drastically reduces the time to create medical training scenarios. The system uses **multiple specialized LLM agents** in a pipeline – e.g. one agent writes patient narratives, another generates vital signs, another formulates debriefing points – all coordinated through a chat interface [19] . Key patterns employed include task *decomposition*, *prompt chaining* outputs between agents, *parallelization* of independent sub-tasks, and iterative refinement with human-in-the-loop review. This practical deployment in an enterprise-like setting (healthcare education) underscores that **agentic workflows yield 70–80% efficiency gains** in content generation without sacrificing quality, a result likely transferable to business proposal generation.	doi:10.1016/j.infoh.2025.03.001
16	**Deep Ideation: Designing LLM Agents to Generate Novel Research Ideas** (Zhao et al., 2025)	4.0	Targets automated **research proposal generation** with a collaborative agent framework. Introduces an *"explore-expand-evolve"* iterative workflow [20] : an LLM agent explores a knowledge graph of scientific concepts, expands ideas by selecting and replacing keywords, and evolves proposals through cycles of critique from a trained Review agent. This **iterative refinement pattern** (with a critic simulating peer review) enabled the system to produce research ideas rated more innovative than those of many human researchers. It highlights that **continuous feedback and revision** – inherent in agentic loops – are vital for producing high-quality, novel proposals.	[arXiv: 2511.02238](arXiv:2511.02238)

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
17	**FlowReasoner: Reinforcing Query-Level Meta-Agents** (Gao et al., 2025)	3.5	Uses **reinforcement learning** to train a *meta-agent* that **assembles a custom multi-agent workflow for each user query**. FlowReasoner's agent learns to choose reasoning strategies and allocate sub-agents (tools vs. debaters, etc.) dynamically per task [21] . This *adaptive assembly* pattern achieved state-of-the-art results on reasoning benchmarks. It suggests that even within the proposal domain, adapting the workflow (number of agents, use of tools, style of interaction) to each proposal's complexity can yield better outcomes than a fixed agentic template, reinforcing the value of **flexible meta-planning** in agentic systems.	[arXiv: 2509.11079]
18	**ScoreFlow: Mastering LLM Agent Workflows via Score-Based Optimization** (Wang et al., 2025)	3.0	Proposes a **continuous optimization** approach to workflow generation. Instead of discrete search, ScoreFlow uses *score-based preference optimization* to adjust a differentiable representation of workflows [22] . By fine-tuning workflows based on a reward signal (success score), it achieved highly effective query-specific agentic plans. The pattern here is treating workflow design as a **learnable function in continuous space**. While technical, it underscores that optimizing proposal-writing workflows can be framed as a machine learning problem, where gradients can fine-tune agent behaviors for higher quality output.	[arXiv: 2502.04306]

Rank	Title (Year)	QA Score	Main Contribution (Agentic Pattern)	URL
19	**RobustFlow: Towards Robust Agentic Workflow Generation** (Xu et al., 2025)	3.0	Investigates the **robustness** of agentic workflows to perturbations in instructions. Finds that existing auto-generated workflows can be brittle – small phrasing changes caused large variations in agent behavior. RobustFlow introduces a training framework with **preference-based fine-tuning** that teaches the workflow generator to produce consistent plans for semantically similar queries [23]. This adds a **stability pattern**: ensuring that the agent's workflow (and proposal content) remains consistent despite reworded requirements or noise. For enterprise use (where input specs may vary by client), this study is a reminder that reliability of agentic AI is as important as raw performance.	arXiv: 2509.21834
20	**CAMEL: Communicative Agents for "Mind" Exploration of Large Language Models** (Li et al., 2023)	3.0	One of the earliest works on multi-agent prompting: proposes a simple yet effective **role-play paradigm** where two LLM agents (a "user" and an "assistant") are assigned roles and communicate to solve a task [24]. By *simulating both sides* of an interaction, CAMEL agents can explore a problem deeply (e.g. brainstorming requirements and then drafting a solution). This demonstrated that even without tools, **two collaborating LLMs** outperform one, confirming the core premise of RQ1 that agentic (interactive) workflows yield better proposals than single-shot prompting in tasks requiring iterative elaboration.	arXiv: 2303.17760

Key Insights: All selected studies underline that as task complexity grows (long documents, multi-modality, precise domain constraints), **agentic AI workflows become not just helpful but often *necessary*** [3] [4]. Single-turn LLM prompts struggle with completeness, reasoning, or adaptability, whereas agentic approaches – whether through multi-agent collaboration, tool integration, or iterative self-improvement – substantially boost performance, reliability, and efficiency in proposal-generation tasks. Essential design patterns (RQ1.1) emerging from this literature include: **role specialization and collaboration** (agents assuming different expert roles) [4], **tool use orchestration** (LLM "planner" agents delegating to external APIs/models) [15], **memory-enhanced reasoning** (storing and revising intermediate results or reflections) [16], **iterative refinement loops** with feedback (either self-critique or a critic agent) [20], and **dynamic workflow adaptation** (the ability to adjust the plan as new info or errors arise) [10]. Crucially, multiple papers show that having **modular, transparent process flows**

(instead of one big black-box prompt) also facilitates human oversight – a practical necessity in enterprise settings like insurance or legal proposal drafting [19].

In conclusion, the SLR finds that agentic AI workflows are often **required for complex proposal generation** (ensuring accuracy, completeness, and efficiency), and it catalogs a toolbox of proven design patterns to realize such workflows. These patterns now form a basis for practitioners to engineer their own quote/proposal-generation agents, combining the strengths of large language models with the robustness of structured, multi-step problem solving.

---

[1] [14] [23] RobustFlow: Towards Robust Agentic Workflow Generation
https://arxiv.org/html/2509.21834v2

[2] Routine: A Structural Planning Framework for LLM Agent System in Enterprise
https://arxiv.org/html/2507.14447v1

[3] Agentic AI: A Comprehensive Survey of Architectures, Applications, and Future Directions
https://arxiv.org/html/2510.25445

[4] [2308.00352] MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework
https://arxiv.org/abs/2308.00352

[5] [2308.08155] AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation
https://arxiv.org/abs/2308.08155

[6] [7] Adaptive Multi-Agent Reasoning via Automated Workflow Generation
https://arxiv.org/html/2507.14393v1

[8] LightAgent: Production-level Open-source Agentic AI Framework
https://arxiv.org/html/2509.09292v1

[9] [10] Flow: A Modular Approach to Automated Agentic Workflow Generation
https://arxiv.org/html/2501.07834v1

[11] [12] [2410.10762] AFlow: Automating Agentic Workflow Generation
https://arxiv.org/abs/2410.10762

[13] [2511.20693] $A^2Flow:$ Automating Agentic Workflow Generation via Self-Adaptive Abstraction Operators
https://arxiv.org/abs/2511.20693

[15] Solving AI Tasks with ChatGPT and its Friends in Hugging Face
https://www.semanticscholar.org/paper/HuggingGPT%3A-Solving-AI-Tasks-with-ChatGPT-and-its-Shen-Song/d1120d67b700e4dfe8b39eb1e48fbdea4e1a0c43

[16] [17] [2303.11366] Reflexion: Language Agents with Verbal Reinforcement Learning
https://arxiv.org/abs/2303.11366

[18] aclanthology.org
https://aclanthology.org/2025.emnlp-main.893.pdf

[19] From prompt to platform: an agentic AI workflow for healthcare simulation scenario design - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC12085049/

[20] Deep Ideation: Designing LLM Agents to Generate Novel Research Ideas on Scientific Concept Network
https://arxiv.org/html/2511.02238v1

[21] Reinforcing Query-Level Meta-Agents | OpenReview
https://openreview.net/forum?id=Tx9HKhGeQW

[22] [2502.04306] ScoreFlow: Mastering LLM Agent Workflows via Score ...
https://arxiv.org/abs/2502.04306

[24] CAMEL: Communicative Agents for "Mind" Exploration of ... - alphaXiv
https://www.alphaxiv.org/overview/2303.17760