

Agentic AI 방법론(AAM) 소개와 제언

· 윤기영 | 한국외국어대학교 겸임교수 | synsaje@gmail.com



서론

인공지능(AI)에 대한 투자는 확대되고 있지만, 많은 기업은 AI 시범사업을 지속적인 비즈니스 가치로 전환하는 데 어려움을 겪고 있다. PoC는 늘어도 핵심 프로세스 전반으로 확장하는 순간 책임 소재, 의사결정 권한, 데이터 흐름, 기존 관리 구조의 불일치가 드러난다. 그 결과 AI 이니셔티브는 고립된 실험에 머물거나 전략 성과에 기여하지 못하고, 때로는 새로운 운영 위험을 만든다. 이는 단순한 기술 통합 문제가 아니라 조직이 복잡성을 어떻게 흡수하고 재배치할 것인가의 문제다.

기존 디지털 전환과 AI 도입 연구는 기술 준비도, 데이터 인프라, 구현 절차를 설명하는 데 기여했지만, AI 전환을 자동화 효율성의 문제로 좁혀 보는 경향이 있었다. 그러나 현대 조직은 고객 기대, 규제, 공급망, 내부 프로세스, 이해관계자의 판단이 서로 얽힌 복잡한 환경에서 운영된다. AI는 업무를 자동화하는 데 그치지 않고 의사결정 흐름과 인간의 인지적 노력 배분을 바꾼다.

본 글은 AI 전환을 조직적 복잡성 관리의 문제로 재정의하고, 이를 위한 방법론적 프레임워크로 Agentic AI Modeling 방법론(AAM)을 제안한다. AAM은 잠재 비즈니스 요구사항 개발, 더블 다이아몬드를 활용한 발산적 문제 정의, 흐름도와 BPMN 2.0 기반 AI 오케스트레이션, 프롬프트 엔지니어링 패턴의 체계화를 결합한다. 목표는 복잡성을 제거하는 것이 아니라 인간, 조직 프로세스, AI 시스템 사이에서 복잡성이 적절한 지점에 배분되도록 설계하는 것이다.

관련 문헌 고찰: 방법론 비교

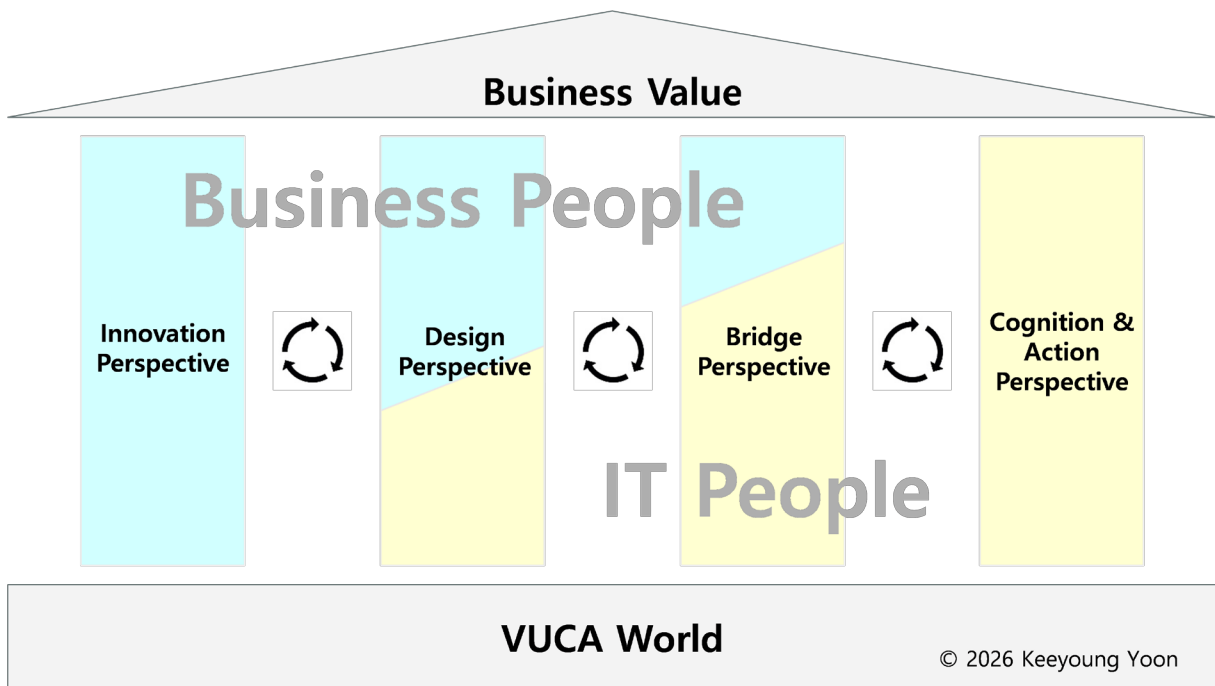
Anthropic, OpenAI, Microsoft, Google Cloud, AWS, IBM 등은 에이전트 설계 패턴, SDK, 런타임, 관측성, 가드레일을 중심으로 Agentic AI 개발 가이드를 제시한다. Deloitte, Accenture, PwC 등 컨설팅 조직은 조직 도입, 운영 모델, 계층형 에이전트 구조, 크로스벤더 오케스트레이션을 강조한다. 이들 접근은 개발과 운영에 유용하지만, 대체로 Agentic AI로 구현할 과제가 이미 정해져 있다는 전제를 둔다.

생성형 AI는 비즈니스와 IT의 경계를 빠르게 좁힌다. 비즈니스 이해관계자의 요구사항은 근본 원인보다 표면적 증상인 경우가 많고, IT 이해관계자는 이를 기술 언어로 번역하는 데 익숙하지만 조직 문제를 깊이 분석하는 훈련은 제한적일 수 있다. 따라서 AAM은 과제 선정 이후의 개발 방법론을 대체하기보다, 과제를 발견하고 문제를 재구성하며 실행 구조로 연결하는 상위 방법론으로 기능한다.

AAM 방법론 개요

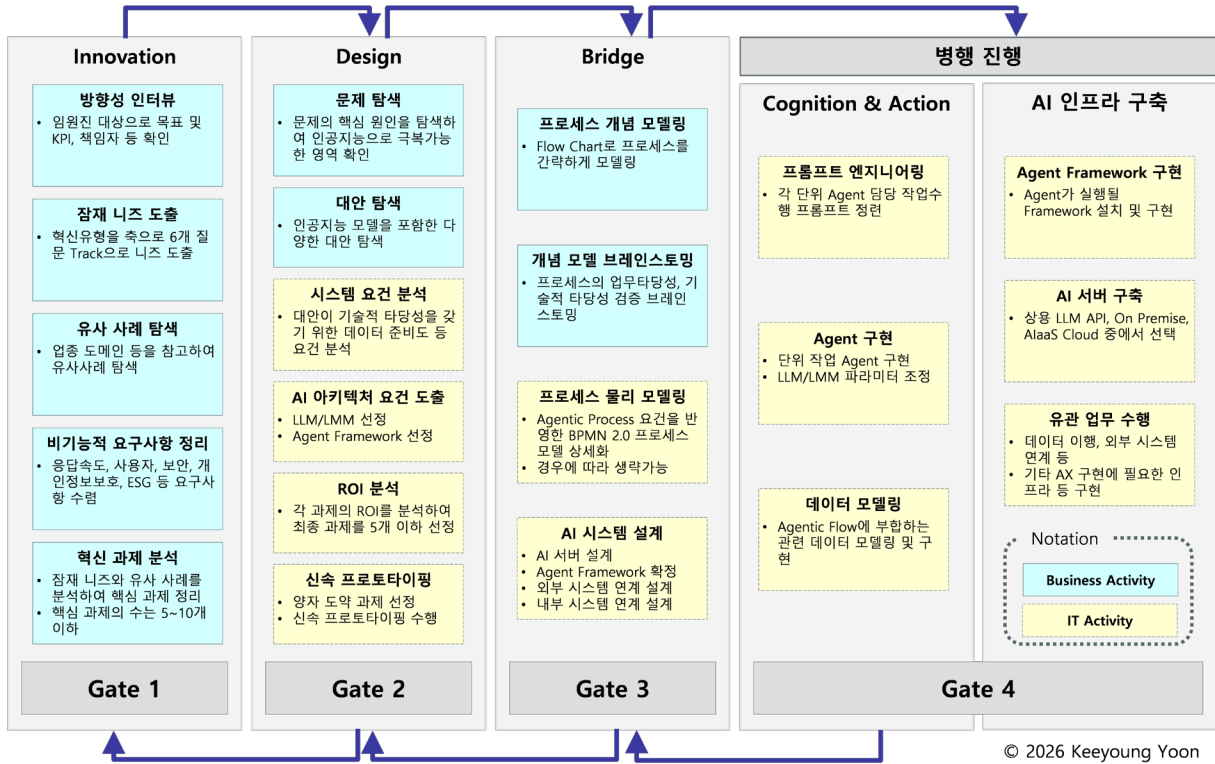
AAM은 특정 벤더 도구나 개발 순서를 규정하지 않는다. 불확실성, 상호의존성, 변화가 큰 조직 환경에서 AI 이니셔티브를 탐색·설계·실행하기 위한 논리를 제공한다. 첫째, 잠재 요구 개발은 기존 요구사항 수집을 넘어 조직이 인식하지 못한 전환 기회를 찾게 한다. 둘째, 발산적 문제 정의는 문제 해석과 해결책 탐색을 성급히 수렴하지 않도록 한다. 셋째, AI 오케스트레이션은 AI 지원 의사결정, 인간 개입, 레거시 시스템, 업무 프로세스의 관계를 명시한다. 넷째, 프롬프트 엔지니어링은 AI의 추론 방식을 조직적으로 표준화하는 인지 인프라가 된다.

[그림 1] AAM 방법론 개념 프레임워크



이 네 요소는 선형 단계라기보다 반복적으로 연결되는 시스템이다. 잠재 요구를 넓게 도출한 뒤 문제를 재해석하고, 선택한 해결책을 프로세스 구조로 바꾸며, 그 안에서 AI가 어떤 방식으로 추론하고 보고하고 중단해야 하는지 설계한다. 운영 중 발견된 위험이나 기회는 다시 요구 개발과 문제 정의 단계로 되돌아간다. AAM의 차별성은 필요성 도출, 설계 탐색, 실행을 하나로 통합하고, 조직 복잡성을 핵심 설계 변수로 다루며, 추상적 원칙을 실행 가능한 산출물과 연결한다는 데 있다.

[그림 2] AAM 방법론 상세 구성



© 2026 Keeyoung Yoon

잠재 비즈니스 요구사항 개발

AI 프로젝트가 실패하는 주요 이유 중 하나는 충분히 구체화되지 않은 요구사항에서 출발하기 때문이다. 전통적 방식은 사용자 인터뷰, 설문, 프로세스 혁신, 유사 사례 분석을 통해 요구를 수집한다. 그러나 복잡한 조직에서 요구는 명확히 주어지지 않는다. 조직 관행, 권력 구조, 데이터 접근성, 고객 경험, 규제 압력, 기존 문제 해석에 의해 형성되며 때로는 말로 표현되지 않는다. 따라서 AAM은 요구를 발견해야 할 고정된 대상이 아니라 적극적으로 개발해야 할 잠재 구성물로 본다.

이를 위해 AAM은 혁신 영역과 질문 경로를 결합한다. 혁신 영역은 비즈니스 및 전략, 고객 경험 및 마케팅, 조직 및 운영, 제품 및 기술, 사람 및 역량을 포함한다. 질문 경로는 고통 완화, 새로운 이익, 모순 파악, 탐색 및 발견, 권한 위임, 연결 및 융합으로 구성된다. 예컨대 “이 업무를 자동화할 수 있는가”가 아니라 “반복적 불편은 무엇인가”, “동시에 만족시키기 어려운 요구는 무엇인가”, “AI에게 맡길 판단과 인간에게 남길 판단은 무엇인가”를 묻는다.

[표 1] 혁신 영역

혁신 영역	핵심 개념
비즈니스 및 전략	기업의 존재 목적을 정의하고 수익 창출 모델을 설계하는 최상위 의사결정 영역. 생태계 및 공급망 혁신, 지속 가능성 혁신 포함
고객 경험 및 마케팅	외부 고객을 향한 가치 전달 방식을 고도화하고, 브랜드 인지 및 상호작용의 질을 극대화하는 영역
조직 및 운영	기획된 전략을 실제 가치로 변환하기 위한 내부 프로세스, 공급망, 조직 구조의 실행 효율성을 극대화하는 영역
제품 및 기술	시장에 제공하는 핵심 산출물의 경쟁력을 강화하고, 이를 뒷받침하는 디지털 기반 기술을 고도화하는 영역
사람 및 역량	모든 혁신을 주도하고 실행하는 주체인 인적 자원을 육성하며, 창의적이고 수용적인 조직 문화를 형성하는 영역

[표 2] 질문 영역

질문 트랙	정의	AX 관점에서의 의미
고통 완화	사용자가 현재 업무나 경험에서 반복적으로 겪는 불편, 부담, 비효율, 스트레스를 줄이기 위한 질문	자동화의 가장 직접적인 출발점으로, AI 적용의 초기 성공 가능성이 높은 영역
새로운 이익	기존에는 얻지 못했거나 인식하지 못했던 새로운 가치, 편익, 기회를 만들어낼 수 있는지 묻는 질문	단순 효율화가 아닌 가치 확장형 AX로 이어지는 핵심 질문
모순 파악	동시에 만족시키기 어려운 상충 요구나 구조적 충돌이 존재하는지 탐색하는 질문	인간이 감당하기 어려운 복합 판단 영역으로, Agentic Process 설계가 특히 효과적인 영역
탐색 및 발견	사용자가 스스로 찾기 어렵거나 놓치고 있는 정보, 패턴, 가능성이 있는지 묻는 질문	LLM 기반 추론과 탐색 능력을 활용해 의사결정의 시야를 확장하는 영역
권한 위임	사람이 직접 하지 않아도 되거나, AI에게 맡길 수 있는 판단·작업이 무엇인지 묻는 질문	자동화 비율을 안정적으로 높이기 위한 핵심 질문으로, 중단·전환 설계와 직결
연결 및 융합	분절된 시스템, 정보, 사람, 업무를 연결하면 새로운 가치가 생기는지 묻는 질문	AX에서 AI를 연결자로 활용하는 대표적인 접근 방식

이 구조는 AI 적용 대상을 눈에 보이는 비효율에 한정하지 않고 비즈니스 모델, 의사결정 권한, 고객 관계, 조직 역량까지 확장한다. 중요한 것은 모든 요구를 AI로 해결해야 한다는 뜻이 아니다. 목적은 어떤 복잡성은 AI 기반 개입에 적합하고, 어떤 복잡성은 인간 판단이나 구조적 통제에 남겨야 하는지 구분하는 데 있다.

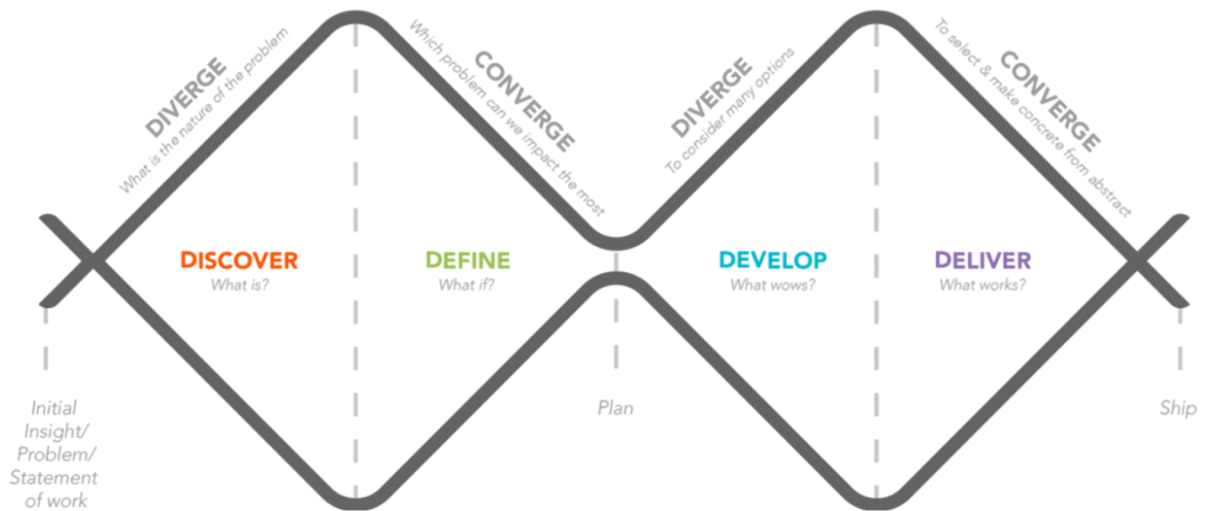
더블 다이아몬드를 활용한 발산적 문제 정의

잠재 요구가 도출되면 조직은 이를 실행 가능한 문제 정의로 바꾸어야 한다. 그러나 많은 AI 이니셔티브는 이 전환을 너무 빠르게 수행한다. 데이터가 있는 문제, 벤더가 제안한 기능, 기존 프로세스의 병목에 맞추어 문제를 조기에 고정하면 조직은 좁은 해석에 갇힌다. AAM은 더블 다이아몬드 프레임워크를 활용해 문제 인식과 해결책 탐색 모두에서 발산을 의도적으로 유지한다.

첫 번째 다이아몬드는 요구의 다양한 해석과 근본 원인을 탐색한다. 효율성 문제로 보이는 요구가 실제로는 권한 배분, 부서 간 조정, 인센티브, 고객 커뮤니케이션 문제일 수 있다. 이 단계에서는 5 Whys, 인과관계 지도, 시스템 사고 등을 활용해 여러 문제 프레임을 병렬로 검토한다. 수렴은 최종 결정이 아니라 작업 가설의 잠정적 안정화로 이해된다.

두 번째 다이아몬드는 해결책 탐색에 초점을 둔다. 여기서도 AI 시스템을 곧바로 답으로 삼지 않는다. 비AI 개입, 조직 재설계, 인간-AI 하이브리드, 데이터 거버넌스 개선 등 여러 경로를 비교한다. 해결책 수렴은 기술 가능성뿐 아니라 조직 복잡성, 책임 구조, 확장성, 운영 위험을 기준으로 이루어져야 한다.

[그림 3] AAM Double Diamond



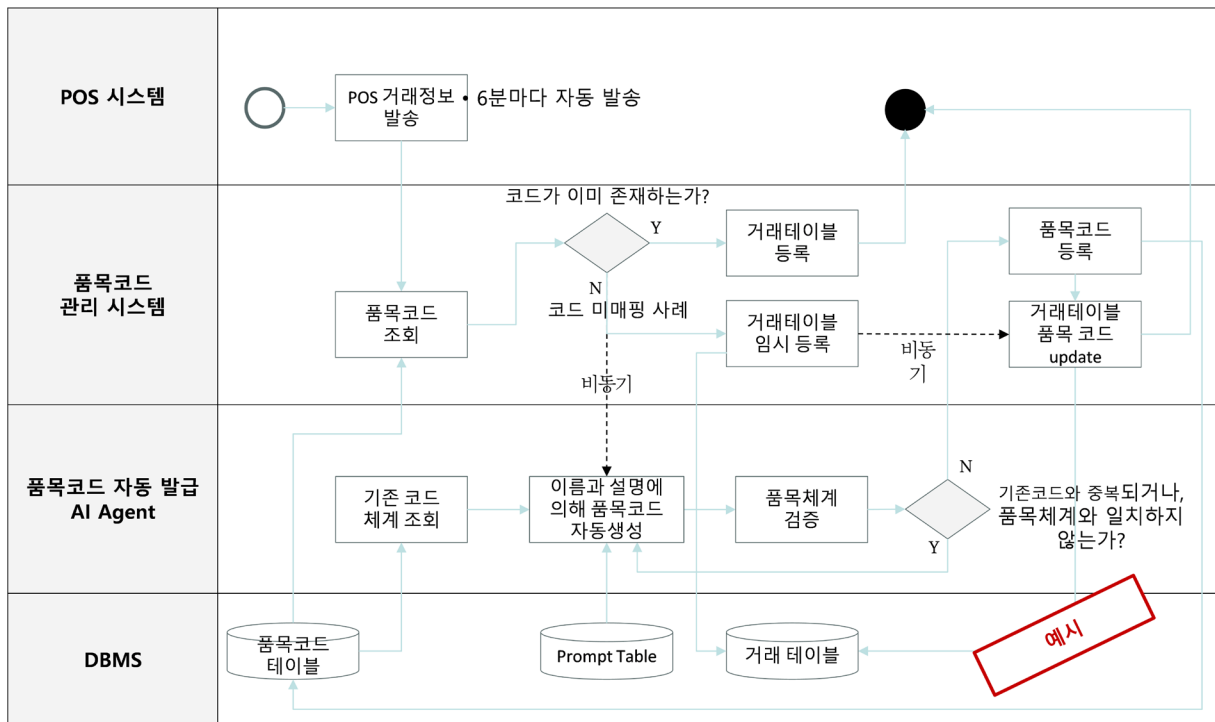
- 대안 도출을 위한 문제는 초기 10개를 넘지 않아야 함
- 근본 원인 분석
- 다중 원인 분석
- 필요시 인과관계지도 작성하여 시스템 구조에서 문제 원인 탐색
- 핵심 니즈와 요구사항 도출
- Machine Learning, Deep Learning, LLM/LMM 등 다양한 대안 탐색
- 필요시 기타 대안도 탐색
- 대안 초안 마련
- 데이터 요건 마련
- 시스템 요건 도출
- 리스크 요인 확인
- 최종 대안은 5개 이내

Bridge: 비즈니스 이해관계자와 IT 이해관계자의 연결

선택된 해결책은 설계 초기에 비즈니스 타당성과 기술 현실성을 동시에 검토해야 한다. 이를 위해서는 비즈니스 이해관계자와 IT 이해관계자가 함께 이해할 수 있는 언어가 필요하다. AAM은 개념 모델과 논리 모델 단계에서 흐름도를 우선 활용할 것을 제안한다. 흐름도는 조직, 사람, 레거시 시스템, AI 에이전트의 역할과 책임을 직관적으로 표현하며, 서로 다른 언어를 사용하는 이해관계자가 같은 그림을 보며 논의하게 한다.

AI 기반 솔루션은 의사결정의 위치, 책임자, 불확실성 처리 방식을 바꾼다. 따라서 AI 오케스트레이션은 기술 파이프라인 설계가 아니라 조직 설계 활동이다. 오케스트레이션은 AI가 어떤 작업을 수행하는가뿐 아니라, 프로세스 안에서 어디에 위치하는지, 산출물이 어떻게 해석되는지, 인간이 언제 개입하는지, 오류와 예외가 어떤 경로로 올라가는지를 정의한다.

[그림 4] AAM Flow Chart 예시



그 다음에 물리 모델링 단계에서는 실제 실행을 위해 BPMN(Business Process Management Notation) 2.0, CMMN(Case Management Model and Notation), DMN(Decision Model and Notation)의 세 가지 도구를 상황에 맞게 쓰는 것으로 고려해야 한다.

BPMN 2.0은 업무 흐름이 정해져 있고 반복되는 경우에 사용한다. 메시지와 이벤트, 예외 처리를 자세히 표현할 수 있어서 엔진에서 바로 실행 가능하다. CMMN은 예외가 많고 경로를 미리 정하기 어려운 경우에 사용한다. 예를 들어 추가 정보가 필요하거나 인간이 개입해야 할 때 이를 유연하게 설계할 수 있는 대안이 된다. DMN은 규칙과 정책을 설계할 때 사용할 수 있다. 승인 기준이나 위험 점수 같은 결정 로직을 표로 정리해서 BPMN이나 CMMN에서 호출한다. 이렇게 하면 복잡한 규칙이 프로세스 그림에 섞이지 않아서 가독성이 좋아진다. 에이전트는 BPMN의 태스크나 CMMN의 단계로 포함시켜서 LLM을 호출한다. 결과적으로 AAM으로 큰 그림을 잡고, 각 도구의 강점을 살려서 실행 가능한 설계를 만들 수 있다.

다만 이런 다양한 도구를 사용하는 것은 초기 진입 문턱을 높인다는 문제가 있을 수 있다. 따라서 이를 사전에 고려하고 논리 모델링과 물리 모델링 설계 정책을 수립해야 한다.

인지 인프라로서의 프롬프트 엔지니어링

오케스트레이션이 AI가 어디에 놓이는지를 정한다면, 프롬프트 엔지니어링은 AI가 어떻게 추론하고 작업을 해석하며 출력을 생성할지를 정한다. 많은 조직은 프롬프트를 개별 사용자의 임기응변에 맡긴다. 이 경우 같은 업무에서도 결과가 일관되지 않고, 추론 과정이 설명되지 않으며, 재사용 가능한 지식이 축적되지 않는다. AAM은 프롬프트를 단순한 입력 문장이 아니라 조직의 의사결정과 인지 구조를 형성하는 설계 산물로 본다.

프롬프트 패턴은 상황에 맞게 선택·조합되어야 한다. 단순 업무에는 Zero-shot, Few-shot, 형식 지정, 역할 부여가 유용하다. 복잡한 판단에는 Chain of Thought, Least-to-Most, Tree of Thoughts, Self-Consistency 처럼 문제를 분해하고 여러 추론 경로를 비교하는 방식이 필요할 수 있다. 최신 정보와 사내 지식이 중요한 경우에는 RAG, Text-to-SQL, Web RAG, Tool-augmented Prompting을 결합한다.

체계화된 프롬프트 설계는 팀이나 개인에 따라 다른 접근을 표준화해 일관성을 높이고, AI 출력의 논리와 검토 지점을 드러내 투명성을 강화하며, 표준 패턴을 여러 프로세스에 재사용해 확장성을 확보한다. 특히 고위험 의사결정에서는 프롬프트가 불확실성을 표시하고, 근거를 분리하며, 인간 검토 조건을 명시하도록 설계되어야 한다.

[표 3] 프롬프트 엔지니어링 대표 패턴

유형	기법	설명
In-Context Learning	Zero-shot Prompting	예시를 제공하지 않고 지시사항(Instruction)만으로 작업을 수행하게 하는 기법
	Few-shot Prompting	소량의 입력-출력 예시를 제공하여 모델이 패턴을 학습하도록 유도하는 기법 (단일 예시인 One-shot 포함)
	Dynamic Few-shot	데이터베이스에서 현재 입력값과 의미적으로 가장 유사한 예시를 동적으로 검색(Retrieval)하여 프롬프트에 삽입하는 기법. 고정된 예시보다 성능이 우수
	Analogical Prompting	모델에게 관련된 예시를 스스로 생성(Self-generate)하게 하고, 이를 바탕으로 본 문제를 해결하게 하는 기법. 적절한 예시 데이터가 없을 때 유용
Reasoning & Logic Enhancement	Chain of Thought (CoT)	문제 해결 과정을 “단계별로 생각하라(Let’s think step by step)”는 지시를 통해 중간 추론 단계를 명시적으로 생성하게 하는 기법
	Contrastive CoT	올바른 추론 과정뿐만 아니라 ‘잘못된 추론’의 예시를 함께 제공하여, 모델이 실수를 피하는 방법을 학습하게 하는 기법
	Least-to-Most Prompting	복잡한 문제를 하위 질문(Sub-questions)들로 분해하고, 쉬운 문제부터 순차적으로 해결하여 최종 답을 도출하는 기법
	Tree of Thoughts (ToT)	여러 개의 추론 경로를 트리 구조로 탐색하고, 각 경로를 평가(Evaluation)하여 최적의 해결책을 찾는 기법
	Graph of Thoughts (GoT)	추론 과정을 네트워크(Graph) 형태로 구조화하여, 서로 다른 생각들을 결합(Aggregation)하거나 순환(Loop)시키는 기법. 가장 복잡한 추론 구조
	Self-Consistency	동일한 질문에 대해 다양한 추론 경로(CoT)를 여러 번 생성한 뒤, 다수결(Majority Vote)을 통해 가장 일관성 있는 답변을 채택하는 기법
	System 2 Attention (S2A)	입력된 텍스트에서 문제 해결에 방해되는 불필요한 정보(Noise)를 먼저 제거하고, 핵심 정보만 남겨 다시 추론하게 하는 기법. 인간의 숙고(System 2) 과정을 모방
Knowledge & Tool Augmentation	Vector RAG	텍스트 임베딩의 유사도를 기반으로 문서를 검색 (비정형 데이터에 적합)
	GraphRAG (Knowledge Graph Prompting)	‘지식 그래프(Knowledge Graph)’의 개체(Entity)와 관계(Relation) 정보를 활용하여 검색. 정보 간의 연결성을 파악하는 복합 추론에 강력
	Text-to-SQL (Structured RAG)	자연어 질문을 SQL 쿼리로 변환하여 정형 데이터베이스(RDBMS)의 정확한 값을 조회
	Web RAG	검색 엔진(Google, Bing 등)을 통해 실시간 인터넷 정보를 검색하여 반영
	Tool-augmented Prompting	LLM이 스스로 해결하기 어려운 문제(복잡한 계산, 최신 정보)를 해결하기 위해 계산기, 코드 인터프리터, API 등 외부 도구를 호출하여 사용하는 기법
	Generated Knowledge Prompting	질문에 답하기 전에 모델이 스스로 관련된 배경지식을 먼저 생성하고, 이를 근거로 답변하게 하여 정확도를 높이는 기법
Structure & Behavioral Control	Role Prompting (Persona)	AI에게 특정 전문가나 역할(Role)을 부여하여 해당 관점과 전문 용어를 사용하게 하는 기법
	Framework-based Prompting	후카츠(Fukatsu), 쉰스케(Shunsuke), CO-STAR 등 검증된 구조적 템플릿을 사용하여 지침, 제약, 입출력을 명확히 정의하는 기법
	Format Specification	답변의 형식을 Markdown, JSON, XML 등으로 강제하여 시스템 통합을 용이하게 하는 기법
	Emotion Prompting	이것은 내 커리어에 매우 중요하다, “자신감을 가져라” 등 인간적인 감정 자극을 프롬프트에 포함하여 모델의 성능 향상을 유도하는 기법
Context & Optimization	Multi-turn Context Injection	긴 대화에서 문맥을 잃지 않도록 이전 대화의 핵심 요약이나 중요 정보를 선별하여 프롬프트에 다시 주입하는 기법
	Prompt Compression	긴 프롬프트를 의미 손실 없이 압축하여 토큰 비용을 절감하고 처리 속도를 높이는 기법
	Reflexion	모델이 생성한 결과물에 대해 스스로 평가하고 수정하는 자기 성찰 루프를 통해 답변 품질을 반복적으로 개선하는 기법
	Active Prompting	모델이 불확실해하는(Perplexity가 높은) 질문을 선별하고, 해당 질문에 대해 인간이 직접 예시(CoT)를 작성하여 학습시키는 기법

논 의

AAM은 AI 전환의 분석 초점을 기술 도입에서 조직 복잡성 관리로 이동시킨다. 기존 연구는 조직 맥락의 중요성을 인정했지만, 복잡성을 핵심 설계 변수로 다루는 데는 한계가 있었다. AAM은 지연된 수렴, 피드백 루프, 적응형 구조 같은 복잡성 원리를 잠재 요구 개발, 더블 다이아몬드, 흐름도 기반 오케스트레이션, 구조화된 프롬프트라는 구체적 활동으로 전환한다. 그 결과 AI 거버넌스는 구현 이후에 덧붙이는 통제가 아니라 전환 과정 자체에 내재된다.

경영 실무 측면에서 AAM은 단편적 사용 사례보다 조직의 전환 요구를 먼저 묻게 하고, 초기 단계에서 특정 AI 솔루션에 성급히 고정되는 것을 피하게 하며, AI 기반 프로세스에서 책임, 검토, 에스컬레이션, 재조정 경로를 명확히 하도록 돕는다. IT 관점에서는 Agentic AI가 전통적 개발 방법론의 수정을 요구한다. 거대언어 모델 기반 시스템은 결과가 확률적으로 변하므로 개발 단계부터 인간 개입 경계, 평가 기준, 모델 표류와 환각 모니터링, 가드레일, 운영 중 재학습 절차를 포함해야 한다.

결론과 향후 과제

AAM은 인공지능 도입을 단순한 기술 구현이 아니라 조직적 복잡성을 관리하고 거버넌스를 확보하는 과정으로 재정의한다. 잠재 요구를 넓게 개발하고, 문제와 해결책을 발산적으로 검토하며, 흐름도와 BPMN 2.0으로 오케스트레이션을 설계하고, 프롬프트 패턴을 인지 인프라로 표준화함으로써 조직은 AI 이니셔티브의 책임성, 견고성, 확장성을 높일 수 있다.

그러나 한계도 분명하다. 확률적·동적 성격을 지닌 Agentic AI를 정적·규칙 기반 표기법인 BPMN 2.0으로 표현하는 데는 논리적 긴장이 있다. 또한 방법론 적용에는 시스템 사고, 메타인지, AX·DX 이해가 필요하므로 조직 전체로 확산하는 데 비용이 든다.

성과를 측정할 정량 지표도 더 정교해져야 한다. 향후 연구는 실제 조직 적용 사례를 통해 AAM이 의사결정 품질, 운영 안정성, 확장성에 미치는 영향을 검증하고, 규제가 강한 산업에 맞게 혁신 영역과 질문 경로를 세분화해야 한다.



참고문헌

- Accenture. (2025). Harnessing the power of AI agents. Accenture. <https://www.accenture.com/us-en/insights/data-ai/hive-mind-harnessing-power-ai-agents>
- Accenture. (2025). The new rules of platform strategy in the age of agentic AI. Accenture. <https://www.accenture.com/us-en/insights/data-ai/platform-strategy-agentic-ai>
- Altshuller, G. (1984). And suddenly the inventor appeared: TRIZ, the theory of inventive problem solving. Technical Innovation Center, Inc.
- Amabile, T. M. (1998). How to kill creativity. *Harvard Business Review*, 76(5), 76-87.
- Amazon Web Services. (2025). Agentic AI patterns and workflows on AWS. AWS Prescriptive Guidance. <https://docs.aws.amazon.com/prescriptive-guidance/latest/agentic-ai-patterns/introduction.html>
- Anthropic. (2024, December 19). Building effective agents. Anthropic. <https://www.anthropic.com/research/building-effective-agents>
- Chesbrough, H. W. (2003). *Open innovation: The new imperative for creating and profiting from technology*. Harvard Business Press.
- Christensen, C. M., Hall, T., Dillon, K., & Duncan, D. S. (2016). *Competing against luck: The story of innovation and customer choice*. HarperBusiness.
- Deloitte. (2024). The cognitive leap: How to reimagine work with AI agents. Deloitte. <https://www.deloitte.com/us/en/what-we-do/capabilities/applied-artificial-intelligence/articles/ai-agents-multiagent-systems.html>
- Design Council. (2026). The Double Diamond: A universally accepted depiction of the design process. <https://www.designcouncil.org.uk/resources/the-double-diamond/>
- Google Cloud. (n.d.). Agent Platform overview. Google Cloud Documentation. <https://docs.cloud.google.com/gemini-enterprise-agent-platform/overview>
- Huyen, C. (2024). *AI Engineering: Building Applications with Foundation Models*. O'Reilly Media, Incorporated.
- IBM. (2025, October 22). Getting started with watsonx Orchestrate Agent Development Kit. IBM Developer. <https://developer.ibm.com/tutorials/getting-started-with-watsonx-orchestrate/>
- IBM. (n.d.). Orchestrating agents. IBM Documentation. <https://www.ibm.com/docs/en/watsonx/watson-orchestrate/base?topic=agents-orchestrating>
- Keeley, L., Pikkell, R., Quinn, B., & Walters, H. (2013). *Ten types of innovation: The discipline of building breakthroughs*. John Wiley & Sons.
- Meadows, D. (2001). Dancing with systems. *Whole Earth*, 106(3), 58-63.
- Meadows, D. (2008). *Thinking in systems: International bestseller*. Chelsea Green Publishing.
- Microsoft. (2025, October 1). Introducing Microsoft Agent Framework: The open-source engine for agentic AI apps. Microsoft Foundry Blog. <https://devblogs.microsoft.com/foundry/introducing-microsoft-agent-framework-the-open-source-engine-for-agentic-ai-apps/>
- Munger, C. T. (2023). *Poor Charlie's Almanack: The essential wit and wisdom of Charles T. Munger*. Stripe Press.

- NANDA, M. (2025). State of AI in business 2025. Preprint at https://www.artificialintelligence-news.com/wp-content/uploads/2025/08/ai_report_2025.pdf.
- OpenAI. (2024). GPT-4.1 Prompting Guide (OpenAI Cookbook).
- OpenAI. (2025). A practical guide to building agents. OpenAI. <https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf>
- OpenAI. (2025, October 6). Introducing AgentKit. OpenAI. <https://openai.com/index/introducing-agentkit/>
- Osterwalder, A., Pigneur, Y., Bernarda, G., & Smith, A. (2014). Value proposition design: How to create products and services customers want. John Wiley & Sons.
- Page, S. E. (2018). The Model Thinker: What You Need to Know to Make Data Work for You. Basic Books
- PwC. (2025, March 27). PwC launches AI agent operating system to revolutionize AI workflows for enterprises. PwC. <https://www.pwc.com/us/en/about-us/newsroom/press-releases/pwc-launches-ai-agent-operating-system-enterprises.html>
- Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. Artificial Intelligence. Prentice-Hall, Englewood Cliffs
- Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927, 1.
- Sasaki, Y., Washizaki, H., Li, J., Yoshioka, N., Ubayashi, N., & Fukazawa, Y. (2025). Landscape and taxonomy of prompt engineering patterns in software engineering. IT Professional, 27(1), 41-49.
- Schumpeter, J. A. (1934). The theory of economic development: An inquiry into profits, capital, credit, interest, and the business cycle. Harvard University Press.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. The knowledge engineering review, 10(2), 115-152
- Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., ... & Gui, T. (2025). The rise and potential of large language model based agents: A survey. Science China Information Sciences, 68(2), 121101