

비개발자를 위한 AI 업무 자동화 판단 보고서

반복 업무를 AI 구조로 바꾸기 전, 무엇을 LLM에 직접 묻고, 무엇을 Claude 프로젝트로 묶고, 어디부터 자동화 도구와 개발 협업이 필요한지 판단하기 위한 실무형 전자책입니다.

목차

00 **핵심 요약**

이 자료에서 가져가야 할 결론과 학습 방향

01 **업무 분류 기준**

자동화 후보, Claude 프로젝트 후보, LLM 직접 사용 후보를 나누는 기준

02 **Claude 프로젝트 구성요소**

CLAUDE.md, reference, command, Skill의 역할 차이

03 **파일 기반 작동 메커니즘**

md 파일과 command 파일이 실제로 작동하는 방식

04 **외부 연결의 현실적 범위**

메일, 게시판, 내부 시스템과 연결될 때의 경계

05 **최종 정리**

강의 마무리용 핵심 메시지

핵심 요약

비개발자가 처음부터 완전 자동화를 목표로 할 필요는 없습니다. 현실적인 1차 목표는 반복 업무의 판단 기준과 문서 양식을 Claude 프로젝트 안에 넣어, AI가 초안 작성과 정리를 안정적으로 수행하게 만드는 것입니다.

- 01 업무 분류는 반복도만으로 판단하지 않습니다. 판단 기준, 입력 자료, 결과물 형식, 리스크, 외부 연결 필요성을 함께 봐야 합니다.
- 02 Claude 프로젝트는 반복 업무를 완전 자동화하기 전, 기준과 참고자료를 모아 AI가 안정적으로 재현하게 만드는 구조입니다.
- 03 command는 자주 쓰는 요청문이고, Skill은 반복 업무 처리 절차입니다. 처음부터 Skill을 만들 필요는 없습니다.
- 04 md 파일은 스스로 실행되는 프로그램이 아닙니다. 사람이 명령하면 AI가 파일을 읽고 지침으로 해석합니다.
- 05 행정 업무에서는 AI가 초안, 요약, 분류를 맡고 최종 판단, 승인, 발송, 책임은 사람이 맡는 Human-in-the-loop 구조가 안전합니다.

업무를 자동화 후보, Claude 프로젝트 후보, LLM 직접 사용 후보로 나누는 기준

결론적으로 업무 분류는 “얼마나 반복되는가”가 아니라, “AI가 안정적으로 처리할 수 있을 만큼 기준과 입력·출력이 구조화되어 있는가”를 보는 문제입니다.

실무에서는 단순히 반복도와 변동성만으로 자동화 여부를 판단하지 않습니다. 어떤 업무는 자주 반복되더라도 판단 기준이 모호하면 자동화하기 어렵고, 반대로 반복 빈도는 낮아도 입력과 출력 형식이 명확하면 Claude 프로젝트로 묶어두는 가치가 있습니다.

따라서 업무를 분류할 때는 오른쪽의 여섯 가지 기준을 빠르게 점검한 뒤, 현재 업무가 LLM 직접 사용, Claude 프로젝트, command, Skill, 자동화 후보 중 어디에 가까운지 판단하면 됩니다.

판단 프레임 6

1. 반복도

같은 흐름이 얼마나 자주 반복되는가

2. 판단 기준

좋은 결과의 기준을 설명할 수 있는가

3. 입력 구조

양식, 예시, 규정이 정리되어 있는가

4. 출력 형식

결과물 형식이 반복적으로 고정되는가

5. 리스크

오류 시 책임과 검토 지점이 큰가

6. 외부 연결

메일, 게시판, 내부 시스템까지 이어지는가

업무 유형별 권장 처리 방식

분류	판단 기준	대표 업무	권장 방식
LLM 직접 사용 후보	일회성 질문이거나 결과 형식이 고정되어 있지 않은 경우	사업계획서 방향 검토, 문장 다듬기, 쟁점 정리, 회의 질문 도출	Claude나 ChatGPT에 직접 질문하고 대화로 조정
Claude 프로젝트 후보	반복되지만 완전 자동화 전 단계이며, 참고자료와 기준을 계속 활용해야 하는 경우	기관 스타일 공문 작성, 사업 공고 요약, 회의록 정리, 민원 답변 초안	CLAUDE.md, reference, 예시 문서, command로 프로젝트 구성
command 후보	같은 작업 지시를 반복해서 사용하는 경우	회의록 요약, 요건표 정리, 보고서 문체 변환	자주 쓰는 프롬프트를 파일로 저장
Skill 후보	작업 절차와 산출물 형식이 어느 정도 정해진 경우	회의록 입력 → 요약 → 액션 아이템 추출 → 보고용 문장 생성	반복 업무 처리 절차 패키지로 구성

분류	판단 기준	대표 업무	권장 방식
자동화 후보	입력, 처리, 출력이 거의 고정되어 있고 사람이 같은 순서로 반복하는 경우	폴더에 들어온 문서 요약, 보고서 취합, 제출서류 누락 체크	Claude Code, 스크립트, n8n, Power Automate 등 활용

실무 적용 체크리스트

체크 질문	판단 의도	예
이 업무는 주 1회 이상 또는 매월 반복되는가?	반복도가 높을수록 프로젝트화 가치가 큼니다.	<input type="checkbox"/>
좋은 결과의 기준을 말로 설명할 수 있는가?	AI에게 맡기려면 판단 기준이 언어화되어야 합니다.	<input type="checkbox"/>
참고할 양식, 예시 문서, 규정이 있는가?	reference로 넣을 자료가 있으면 Claude 프로젝트에 적합합니다.	<input type="checkbox"/>
결과물 형식이 매번 비슷한가?	출력 형식이 고정될수록 command나 Skill로 확장하기 좋습니다.	<input type="checkbox"/>
개인정보, 예산, 결제, 외부 발송과 관련 있는가?	리스크가 있으면 자동화보다 사람 검토 지점을 먼저 설계해야 합니다.	<input type="checkbox"/>
내부 시스템 로그인, 권한, API 연결이 필요한가?	외부 연결이 필요하면 비개발자 단독 자동화 범위를 넘을 수 있습니다.	<input type="checkbox"/>

해석 기준: 예가 1~2개면 LLM 직접 사용, 3~4개면 Claude 프로젝트 후보, 5개 이상이면 자동화 후보일 수 있습니다. 단, 리스크 항목에 체크가 있다면 사람 검토 지점을 먼저 설계해야 합니다.

실무 분류 예시

LLM 직접 사용

기획 방향 검토

결과가 매번 다르고 판단 비중이 크기 때문에 프로젝트화보다 직접 대화가 적합합니다.

Claude 프로젝트

공문 초안 작성

문체, 포함 항목, 제외 표현, 수정 포인트가 정리되면 프로젝트화하기 좋습니다.

자동화 후보

폴더 문서 요약

입력 폴더, 처리 방식, 출력 폴더가 고정되면 자동화 도구를 활용할 수 있습니다.

CLAUDE.md, reference, command, Skill의 역할 차이

CLAUDE.md는 프로젝트의 기본 규칙, reference는 참고자료, command는 저장해둔 요청문, Skill은 업무 매뉴얼과 실행 절차에 가깝습니다.

Claude 프로젝트를 처음 구성할 때 가장 많이 헷갈리는 부분은 command와 Skill의 차이입니다. command는 사람이 반복해서 입력하는 요청문을 파일로 저장한 것입니다. 반면 Skill은 단순히 “무엇을 해줘”가 아니라, 업무를 어떤 순서로 처리하고 어떤 기준으로 판단할지까지 포함합니다.

구성요소 역할 맵

구성요소	역할	쉽게 말하면	예시
CLAUDE.md	AI가 따라야 할 기본 운영 규칙	프로젝트 사용설명서	문체, 금지사항, 출력 순서, 확인 필요 기준
reference	AI가 참고해야 할 자료	교재, 예시, 규정집	기존 공문 샘플, 보고서 양식, 내부 매뉴얼
command	자주 쓰는 요청문	저장해둔 프롬프트	회의록 요약, 요건표 정리, 문체 변환
Skill	반복 업무 처리 방식	업무 매뉴얼 + 실행 절차	안건 분리, 결정사항 추출, 담당자/기한 표시

command와 Skill의 차이

구분	command	Skill
핵심 질문	무엇을 해줘?	어떻게 처리할까?
성격	짧고 직접적인 요청문	작업 절차와 판단 기준을 포함한 처리 방식
예시	이 회의록을 요약해줘. 이 공고문을 요건표로 정리해줘.	회의 목적 파악 → 안건 분리 → 결정사항 추출 → 담당자와 마감일 정리 → 확인 필요 표시
사용 시점	반복 요청을 줄이고 싶을 때	반복 업무의 처리 방식이 안정적으로 고정되었을 때

처음 시작할 때 권장 순서

STEP 1

Claude 프로젝트 만들기

회의록 정리, 공문 초안 작성, 사업공고 요약처럼 하나의 업무 단위로 시작합니다.

STEP 2

reference 넣기

기존 문서, 양식, 예시, 표현집, 규정 문서를 넣습니다.

STEP 3

CLAUDE.md 작성

문체, 금지사항, 출력 순서, 확인 필요 기준을 정리합니다.

STEP 4

command 2~3개 작성

자주 쓰는 요청부터 저장하고, 반복이 안정되면 Skill로 확장합니다.

비개발자 기준에서는 처음부터 Skill까지 만들 필요가 없습니다. 먼저 프로젝트와 reference, CLAUDE.md, command 몇 개로 시작한 뒤, 같은 수정이 반복되거나 절차가 고정되면 그때 Skill로 확장하는 방식이 가장 안전합니다.

md 파일과 command 파일이 실제로 작동하는 방식

md 파일은 프로그램이라기보다 AI가 읽는 업무 매뉴얼입니다. 사람이 “이 매뉴얼대로 처리해줘”라고 요청할 때 작동합니다.

md 파일이나 command 파일은 그 자체로 실행되는 프로그램이 아닙니다. 사람이 명령을 내리거나, Claude Code 같은 도구가 특정 파일을 읽도록 요청해야 작동합니다. 비개발자 관점에서는 “파일이 실행된다”기보다 “AI가 파일을 읽고 그 안의 규칙을 작업 지시로 해석한다”고 이해하는 것이 정확합니다.

회의록 요약 프로젝트의 폴더 구조

```
meeting-summary-project/  
├─ CLAUDE.md  
├─ reference/  
│   └─ meeting_format.md  
├─ commands/  
│   └─ summarize_meeting.md  
├─ input/  
│   └─ 2026-05-06_meeting.md  
└─ output/
```

작동 순서

STEP 1

입력 파일 배치

사람이 회의록 파일을 input 폴더에 넣습니다.

STEP 2

명령 입력

commands/
summarize_meeting.md
지침에 따라 요약해달라고 요청합니다.

STEP 3

AI가 순서대로 읽음

CLAUDE.md →
command → reference
→ input 순서로 해석합니다.

STEP 4

결과 저장

규칙과 형식에 맞춰
output 폴더에 결과 파일을 저장합니다.

사람과 AI의 역할 분리

단계	사람이 하는 일	AI가 하는 일
		아직 작업하지 않습니다. 읽을 대상이 준비된 상태입니다.

단계	사람이 하는 일	AI가 하는 일
1. 입력 준비	회의록, 공문, 보고서 등 처리할 파일을 input 폴더에 넣습니다.	
2. 작업 요청	command 파일 또는 직접 문장으로 “이 지침대로 처리해줘”라고 명령합니다.	이번 작업의 목적과 처리 범위를 파악합니다.
3. 지침 해석	필요하면 참고자료나 출력 형식을 추가로 지정합니다.	CLAUDE.md, command, reference, input 파일을 읽고 규칙을 조합합니다.
4. 결과 확인	output 결과물을 검토하고, 오류나 누락 정보를 확인합니다.	정해진 형식에 맞춰 결과를 작성하고 output 폴더에 저장합니다.

AI가 읽는 파일별 역할

파일	역할	예시 내용
CLAUDE.md	프로젝트 전체 규칙	결정사항과 논의사항 구분, 공공기관 보고용 문체, 확인 필요 표시
reference/meeting_format.md	결과물 형식 예시	회의 개요, 주요 논의사항, 결정사항, 액션아이템, 확인 필요 사항
commands/summarize_meeting.md	이번 작업 요청문	input 폴더의 회의록을 읽고 reference 형식에 맞춰 output에 저장
input 파일	실제 처리 대상	회의 원문, 담당자, 일정, 논의 내용
output 폴더	결과 저장 위치	회의 요약본, 액션아이템 표, 확인 필요 사항

메일, 내부 게시판, 기관 시스템과 연결되는 업무의 현실적 범위

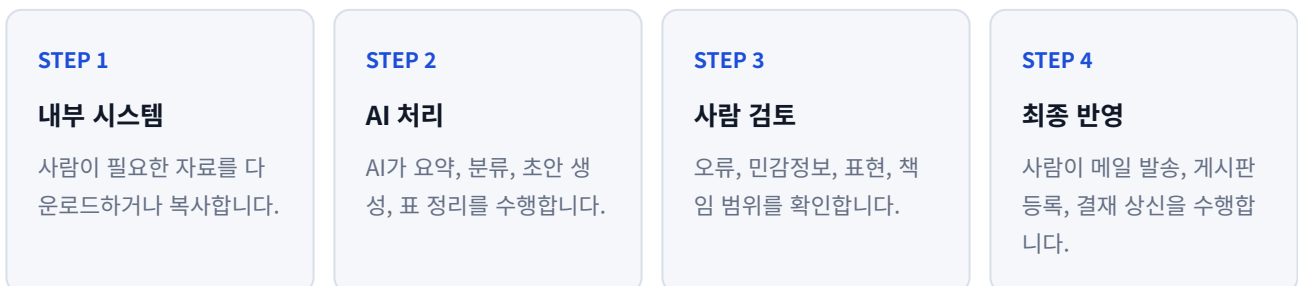
행정 업무에서 가장 안전한 설계는 AI가 초안·요약·분류를 맡고, 최종 판단·승인·발송·등록은 사람이 맡는 Human-in-the-loop 반자동화입니다.

행정 업무는 메일, 내부 게시판, 그룹웨어, 전자결재, 내부 시스템과 연결되는 경우가 많습니다. 하지만 비개발자가 처음부터 모든 시스템을 자동 연결하려고 하면 보안, 권한, 책임 문제가 발생할 수 있습니다. 따라서 외부 연결은 단계적으로 접근해야 합니다.

외부 연결의 현실적 범위

단계	범위	설명
1단계	파일 기반 자동화	문서 요약, 회의록 정리, 공문 초안 작성, 보고서 문체 변환, 제출서류 체크리스트 생성 등. 내 PC나 프로젝트 폴더 안의 파일을 대상으로 하므로 비개발자도 충분히 가능합니다.
2단계	반자동화	AI가 메일 초안이나 공지문, 보고서 요약을 만들고 사람이 Outlook, Gmail, 내부 게시판, 결재 시스템에 반영하는 방식입니다. 행정 실무에서 가장 현실적인 형태입니다.
3단계	외부 도구 연동	Google Apps Script, Zapier, Make, n8n, Slack Workflow, Notion Automation, Microsoft Power Automate 등을 활용할 수 있습니다. 다만 기관 내부망이나 보안 정책에 따라 제한될 수 있습니다.
4단계	개발 협업 필요 영역	내부 시스템 로그인 자동화, 전자결재 자동 상신, 개인정보 포함 데이터 처리, 메일 자동 발송, 외부 API 연동, 내부망 시스템 직접 연결은 개발자·보안 담당자·시스템 담당자와 협업해야 합니다.

권장 운영 구조



가능 / 주의 / 제한 판단표

구분	업무 예시	권장 처리 방식
가능	문서 요약, 회의록 정리, 보고서 초안, 제출서류 체크리스트 생성	AI 활용 가능. 단, 최종 제출 전 사람 검토는 유지합니다.
주의	메일 초안, 게시판 공지문, 민원 답변 초안, 결재 문서 초안	시가 초안을 만들고 사람이 표현, 사실관계, 책임 범위를 검토한 뒤 반영합니다.
제한 / 협업 필요	자동 발송, 전자결재 자동 상신, 개인정보 처리, 예산·계약·인사 판단, 내부 시스템 직접 입력	비개발자 단독 자동화는 피하고, 개발자·보안 담당자·시스템 담당자와 협업합니다.

외부 연결 판단 질문

- 로그인이나 권한이 필요한가?
- 개인정보나 민감정보가 포함되는가?
- 결과가 외부로 발송되거나 등록되는가?
- 실패했을 때 책임 문제가 생기는가?
- 공식 API나 안정적인 연동 방식이 있는가?

위 질문 중 하나라도 해당된다면 완전 자동화보다 사람 검토 지점을 먼저 설계해야 합니다. 공식 API가 있고 보안 검토를 통과한 경우에는 개발 협업으로 안정적인 연결을 만들 수 있지만, 공식 연동이 없고 화면을 억지로 조작해야 한다면 위험도가 높습니다.

CONCLUSION

최종 결론

- 비개발자가 처음부터 완전 자동화를 목표로 할 필요는 없습니다.
- 현실적인 1차 목표는 반복 업무의 판단 기준과 문서 양식을 Claude 프로젝트 안에 넣는 것입니다.
- AI는 초안, 요약, 분류, 정리를 맡고 최종 판단, 승인, 발송, 책임은 사람이 맡아야 합니다.
- 외부 시스템 연결은 자동화의 마지막 단계이며, 그 전까지는 반자동화 구조가 가장 안전합니다.

한 줄로 정리하면, 처음 목표는 “모든 것을 자동화하는 것”이 아니라 내 업무 기준을 Claude가 흔들림 없이 재현하게 만드는 것입니다.