

1 — Три примитива (дайте модели все три — и она станет агентом)

Инструменты	Дотянуться до мира за пределами текста: детектор, расшифровка, запрос в архив.
Память	Хранить и вспоминать факты между шагами. Окно — это лишь рабочая память.
Планирование	Превратить цель в шаги; разбить большую задачу на мелкие действия.

2 — Анатомия вызова инструмента (function calling)

- ОПИСАТЬ инструмент = имя + описание + JSON Schema входов
 - ЗАПОЛНИТЬ модель выбирает инструмент и заполняет аргументы (НЕ запускает)
 - ЗАПУСТИТЬ ваша среда проверяет анкету и запускает настоящую функцию
 - НАБЛЮДАТЬ результат возвращается модели как следующее наблюдение
- MCP один стандарт подключает инструменты: обнаружение, схема, вызов (2024)

3 — Четыре вида памяти (в окно влезает только рабочая память)

Рабочая	текущая задача + шаги	-> контекстное окно	(только эта задача)
Эпизодическая	прошлые события	-> внешнее хранилище	(между сессиями)
Семантическая	общие факты и знания	-> поисковая база	(долгоживущая)
Процедурная	как делать задачу	-> промпт / веса	(постоянная)

4 — Три паттерна планирования (интерактив / повторяемо / ответственно)

ReAct	думать -> действовать -> наблюдать	интерактив (расследование)
Plan-and-execute	план заранее, затем исполнение	повторяемо (ночной пакет)
Reflection	выдать -> самокритика -> повтор	ответственно (отчёт)

5 — Оставь простым: добавляй примитив только когда задача требует

- ☐ Влезает в одно окно? Не нужна внешняя память — не разворачивайте vector DB.
- ☐ Шаги не меняются? Запишите их кодом (workflow), а не открытым планировщиком.
- ☐ Один сервис один раз? Дайте агенту один инструмент, а не фреймворк инструментов.
- ☐ Много инструментов вредит: больше инструментов = хуже выбор. Берите наименьший набор.
- ☐ Не переполняйте окно: платите за каждый токен и хороните важные факты.