

Память AI-агентов: архитектура и практическая настройка

Часть 1

Конспект к записи о том, из каких контуров складывается память агента и почему надёжная работа невозможна без правил, задач, навыков и внешних источников знаний.

«Память агента — это не один диалог. Это согласованная система, которая помогает сохранять контекст, правила и знания проекта».

Зачем смотреть эту запись

Понять архитектуру

Разобраться, чем рабочий контекст отличается от ролей, навыков, проектных документов и базы знаний.

Снизить ошибки

Увидеть, почему агент теряет смысл задачи, начинает фантазировать или выходит за границы полномочий.

Собрать основу

Понять место файлов правил, skills, Kanban, Obsidian, истории сессий и RAG в одной системе.

Подготовиться к практике

Перейти от общей схемы к настройке памяти и проверке её качества на следующей встрече.

Карта записи

00:00:00–00:09:15

Рамка цикла и контуры памяти. Память агента рассматривается как набор взаимосвязанных слоёв, а не как единственный чат.

00:09:15–00:19:42

Контекст LLM и сессии. Сжатие токенов, исторические данные, пересечение сессий и риск утраты важного контекста.

00:19:42–00:38:30

Рабочая память, роли и Kanban. Как агент хранит ход задачи, что ему можно и нельзя, и где живут статусы работы.

00:38:30–01:10:02

Память проекта и мультиагентная система. Obsidian, проверка результата, история сессий, кэш и агентная оболочка.

01:10:02–01:41:51

Skills, профили и внешний слой знаний. Процедурная память, Agent MD / Soul MD / Memory MD, RAG и ответы на вопросы участников.

1. Рабочая память: что происходит прямо сейчас

Рабочая память удерживает задачу, недавние сообщения и результаты инструментов в пределах текущей сессии. Важно не путать историю чата, которую видит человек, с тем контекстом, который доступен модели в конкретный момент.

1 Сессия имеет границы. В Hermes обсуждаются обновления по времени: в том числе 24-часовой цикл и настройка перехода в 04:00.
00:19:42–00:24:42

2 Контекст может сжиматься. При сжатии система сама решает, что оставить; поэтому значимые для задачи сведения нужно закреплять в подходящем контуре памяти.
00:13:27–00:17:03

3 Кэш — часть работы, а не мусор. В записи кэш рассматривается как полезный механизм сохранения и извлечения контекста; постоянная ручная чистка не рекомендуется.
01:05:02–01:07:07

2. Роли, правила и границы

Агенту недостаточно знать, как выполнить задачу. Ему нужно помнить, **кем он является, какие у него полномочия и какие действия запрещены**. Когда этот слой пуст или теряется, поведение становится непредсказуемым.

- Роль описывает назначение и ожидаемый способ взаимодействия.
- Правила фиксируют обязательные процедуры и требования к качеству.
- Границы определяют, что агенту доступно, а что требует человека или иной роли.

00:24:42–00:27:12

3. Процедурная память: skills

Skills в записи объясняются через библиотечную картотеку: агент не перебирает все инструкции подряд, а находит нужную карточку и применяет конкретную процедуру к конкретной задаче.

Что даёт skill

Повторяемый способ выполнять действие: от запуска инструмента до проверки результата.

Что важно контролировать

Неправильный или устаревший skill может вести по неверному маршруту.

Почему нужны ограничения

В командном контуре нельзя позволять разным участникам бесконтрольно переписывать общую процедуру.

Как улучшать

Изменения должны быть осознанными, проверяемыми и привязанными к конкретной задаче улучшения.

01:10:02–01:18:18

4. Операционная память: задачи и ответственность

Kanban в этой архитектуре — не просто список дел. Он помогает агенту и человеку видеть:

- что выполняется сейчас;
- что ожидает проверки;
- кто отвечает за следующий шаг;
- какие действия должны повторяться по расписанию.

Это делает работу воспроизводимой и снижает риск потерять задачу при смене сессии или исполнителя.

00:28:26–00:29:35, 01:26:50–01:29:59

5. Память проекта: документы, решения и источник истины

Проектная память нужна, чтобы знания команды не исчезали вместе с текущим чатом, моделью или отдельным агентом. В записи эту роль выполняет Obsidian: там сохраняются документы, решения и контекст работы.

Память проекта должна переживать смену чата, модели и агента.

Что хранить в человекочитаемом контуре

- правила проекта и регламенты;
- контекст продуктов и процессов;
- принятые решения и результаты проверок;
- ответственность и статусы задач;
- полезные инструкции, к которым агент может вернуться.

00:48:29–00:55:22, 01:31:50–01:37:26

Obsidian и RAG — не одно и то же

Obsidian удобен как общая, читаемая человеком и агентом память проекта. Векторная база и RAG — другой слой: он предназначен для семантического поиска и может выступать внешним источником истины. Эти контуры дополняют друг друга.

01:34:42–01:37:26

6. Из чего складывается устойчивый агент

Идентичность и правила

Agent MD, Soul MD и Memory MD связывают назначение агента, границы, знания и способ работы.

01:23:30–01:26:24

Рабочий контекст

Сессия, история, кэш и результаты инструментов позволяют продолжать задачу без потери нити.

00:19:42–00:24:42, 01:05:02–01:08:45

Процедуры

Skills превращают повторяемую работу в проверяемый маршрут, а не в импровизацию каждый раз.

01:10:02–01:18:18

Память проекта и знания

Obsidian, Kanban, базы данных и RAG удерживают накопленный опыт и поддерживают принятие решений.

00:48:29–00:55:22, 01:34:42–01:37:26

Что продолжится на следующей встрече

- Практическое устройство памяти агента и её настройка.
- Ошибки в файлах и контурах памяти, которые влияют на поведение агента.
- Внешняя долгосрочная память: базы данных, RAG и выбор сервисов.
- Практика работы с кэшем и улучшением памяти.

00:01:13–00:02:28, 01:00:46–01:01:07, 01:06:36–01:07:07