

СЕРГЕЙ ЛЕВЧЕНКО

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ПЕДАГОГИКЕ

Практическая книга для учителя

Инструмент. Границы. Решение учителя.

Сергей Левченко

**Искусственный
интеллект в педагогике**

«Автор»

2026

Левченко С.

Искусственный интеллект в педагогике / С. Левченко — «Автор», 2026

Нейросеть за секунды пишет конспект, предлагает задания, перерабатывает материалы и формулирует обратную связь. Но кто проверит её — и где заканчивается помощь и начинается решение, которое учитель должен принимать сам? «Искусственный интеллект в педагогике» — практическая книга для учителя, который хочет использовать ИИ спокойно, системно и без технологического восторга. Здесь нет набора «волшебных промтов» и обещаний освободить половину рабочего дня. Сергей Левченко показывает, как собрать рабочий контур: определить правила, передать системе нужный контекст, разделить роли, проверять факты и методику, проектировать задания, анализировать ошибки, учитывать ограничения конкретного класса и сохранять контроль над результатом. Эта книга не предлагает заменить учителя технологией. Она помогает понять, какую операцию можно передать системе, что необходимо проверить и где ответственность всегда остаётся у человека. Инструмент. Границы. Решение учителя.

© Левченко С., 2026

© Автор, 2026

Содержание

Глава 1. Чат, который ничего вам не должен	5
Глава 2. Как машина продолжает фразу	8
Глава 3. Ошибаться полным голосом	11
Глава 4. Почему хороший запрос не спасает плохую задачу	14
Глава 5. Учебная цель появляется раньше материала	17
Глава 6. Что считается доказательством понимания	20
Глава 7. Где заканчивается подготовка и начинается решение	23
Глава 8. Какую работу стоит отдавать машине	26
Глава 9. Ядро начинается с запрета	29
Глава 10. Паспорт учителя — не автобиография	33
Конец ознакомительного фрагмента.	34

Сергей Левченко

Искусственный интеллект в педагогике

Глава 1. Чат, который ничего вам не должен

Учитель открывает нейросеть обычно не ради знакомства с технологией. Ему нужен результат. До урока осталось сорок минут, в восьмом классе провалилась тема, администрация попросила представить план внеурочного занятия, а дома лежит пачка работ, к которой не хочется подходить. В пустом поле появляется курсор. Человек пишет: «Составь урок по теме...»

Через несколько секунд на экране возникает аккуратный текст. Цель, задачи, ход занятия, вопросы, рефлексия. Всё разложено по местам. Ничего не забыто. Иногда результат выглядит лучше старого конспекта, который много лет переезжал из папки в папку. Возникает почти физическое облегчение: наконец-то рядом появился помощник.

Именно здесь начинается первая ошибка.

Обычный диалог с нейросетью ещё не является помощником. Система отвечает на полученное задание в пределах того контекста, который ей доступен. Она не знает, какой учебник лежит на партах, что класс проходил на прошлой неделе, почему тема не усвоилась, насколько жёстко в школе относятся к формулировкам рабочих программ и какой вопрос директор задаст первым, если эти сведения ей не передали. Она не обязана угадывать даже то, что учитель считает само собой разумеющимся.

Человек видит завершённый документ и невольно достраивает за ним профессиональную работу. Кажется, будто кто-то сначала понял задачу, сверился с программой, выбрал методический подход, проверил факты, оценил время и только потом написал текст. На деле все эти этапы могли не состояться. На экране есть результат, но за ним ещё нет процесса, которому можно доверять.

Разница принципиальная. Конспект можно получить случайно удачным. Рабочая система должна давать приемлемый результат повторно, на разных темах и в разных условиях. Она должна останавливаться, когда не хватает данных, показывать сомнительные места, отделять факт от предположения и не переходить границы, которые установил учитель. Чат этого не делает сам по себе.

Представим вполне обычную задачу: подготовить урок по новой теме для шестого класса. В запросе указаны предмет и название темы. Нейросеть выдаёт структуру на сорок пять минут, добавляет групповую работу и домашнее задание. Формально просьба выполнена. Но неизвестно, соответствует ли материал конкретной федеральной рабочей программе, какие понятия уже введены, есть ли в кабинете нужное оборудование, сколько учеников способны работать в группе без потери половины урока и не требует ли тема предварительного повторения. Красивый план может оказаться педагогически пустым. Он похож на урок, но не продолжает обучение данного класса.

У человека такая ошибка обычно заметна по поведению. Незнакомый коллега спросит, уточнит, нахмурится, полезет в документы. У нейросети нет обязательного внешнего признака сомнения: неверный ответ она способна написать тем же ровным и уверенным языком, что и верный. Форма выглядит как результат внутренней проверки, хотя сама по себе ничего не доказывает.

Поэтому начинать нужно с трезвого определения.

Нейросеть — это не цифровой методист, не второй учитель и не человек, которому достаточно один раз объяснить ситуацию. Это вычислительная система, способная работать с

языком и создавать связанные ответы на основе полученного контекста и усвоенных закономерностей. Она может выполнять многие действия, похожие на профессиональные: сравнивать, перерабатывать, группировать, объяснять, предлагать варианты. Сходство результата с человеческой работой бывает впечатляющим. Из этого не следует, что внутри возник тот же путь к результату.

Для практики важнее другое: нейросетью можно управлять. Не полностью и не безошибочно, но достаточно, чтобы превратить случайный диалог в устойчивый рабочий контур. Для этого учитель должен перестать относиться к ней как к готовому сотруднику и начать собирать сотрудника из правил.

У настоящего помощника есть должность. Он понимает, за что отвечает, какие материалы использует, что должен вернуть, когда обязан остановиться и кому передаёт работу дальше. У него есть доступ только к тем данным, которые нужны. Его результат проверяет другой человек или отдельный этап процесса. Если задача меняется, меняется и маршрут.

В обычном чате всё смешано. В одном диалоге нейросеть успевает быть автором задания, методистом, корректором, экспертом по праву и судьёй ученической работы. Она проверяет собственный текст собственными средствами, а затем уверяет, что всё верно. Это удобно, пока ошибка не имеет последствий. В школе последствия появляются быстро: неверный ответ попадает в контрольную, расплывчатый критерий — в оценивание, чужая формулировка — в характеристику ребёнка, непроверенная норма — в служебный документ.

Система начинается с разделения обязанностей.

Один режим помогает сформулировать учебную цель. Другой создаёт материал только после того, как цель определена. Третий ищет фактические и методические слабости. Четвёртый проверяет, можно ли выполнить задание без понимания темы. Результат не переходит дальше автоматически: учитель видит, что произошло на каждом этапе, и может вернуть работу назад.

Все эти режимы способны работать внутри одной нейросети. Не нужны отдельные программы с громкими названиями. На первом уровне достаточно нескольких диалогов и обычных текстовых файлов. Сложность здесь не техническая. Нужно научиться видеть собственную работу как последовательность решений, а не как одну просьбу: «Сделай мне урок».

Такое изменение поначалу раздражает. Казалось бы, технология обещала экономию времени, а книга предлагает описывать правила, готовить источники, разделять задачи и проверять результат. Почему просто не написать подробный запрос?

Потому что подробный запрос решает только одну задачу в одном разговоре. Архитектура сохраняет способ работы. Однажды настроенный модуль можно использовать для новой темы, другого класса и следующего учебного года. Его можно исправить, если он ошибается, перенести в другой сервис, отключить, если он больше не нужен. Учитель перестаёт зависеть от удачи и от собственной способности каждый вечер заново сочинять идеальное обращение к машине.

Здесь проходит первая важная граница. Нейросеть должна получать ровно те сведения, которые нужны для текущей работы. Если в диалог без порядка навалены переписка, документы, старые инструкции и новые условия, становится трудно понять, что система действительно учла, что пропустила, а что связала неверно. Большой объём контекста ещё не создаёт память, методическую базу и контроль. Их приходится проектировать отдельно.

В дальнейшем мы соберём ядро правил, паспорта курса и класса, библиотеку источников, рабочие модули и проверяющий контур. Но до сборки нужно разобраться с самой машиной. Не на уровне формул и программирования. Учителю не требуется становиться разработчиком языковых моделей. Ему требуется понимать несколько вещей, без которых любое управление остаётся имитацией.

Как появляется ответ? Что система видит в нашем тексте? Почему она может правильно разобрать сложную задачу и тут же выдумать название документа? Что происходит, когда разговор становится длинным? Почему вежливая просьба «проверь ещё раз» иногда лишь делает ошибку убедительнее?

Пока этих ответов нет, чат действует почти гипнотически. Белое поле, бегущие строки, ровный голос. Очень легко принять удобство за надёжность.

Учитель закрывает готовый конспект и открывает программу. Тема стоит в другом разделе. Это небольшая ошибка — урок ещё не начался. Но именно с таких несовпадений обычно и выясняется, что помощника пока никто не собирал.

Глава 2. Как машина продолжает фразу

Разговор о нейросетях часто ломается на двух крайностях. В одном случае систему описывают как почти живой разум, который читает, понимает и рассуждает. В другом — как механический автомат, бездумно подставляющий слова. Обе картины удобны, обе быстро начинают мешать.

Учителю не требуется решать философский вопрос о природе машинного мышления. Для работы достаточно практической модели: языковая система получает последовательность токенов, учитывает доступный контекст и шаг за шагом создаёт ответ. На каждом шаге она рассчитывает, какое продолжение лучше согласуется с тем, что уже дано и создано. Затем новый токен входит в контекст, и расчёт повторяется.

Эти элементы называют токенами. Токен не всегда совпадает со словом: им может быть короткое слово целиком, часть длинного слова, знак препинания, число или фрагмент записи. Пользователю почти никогда не нужно считать токены вручную. Важно другое: для модели текст существует как последовательность закодированных элементов, а не как готовая мысль, которую она одним взглядом увидела целиком.

Отсюда следует первое практическое последствие. Для машины формулировка — это не прозрачная оболочка готового смысла. Порядок слов, соседние инструкции, примеры, названия разделов и даже оформление влияют на то, какое продолжение станет вероятнее. Поэтому два запроса, которые человеку кажутся почти одинаковыми, могут дать заметно разные ответы.

Во время предварительного обучения языковая модель обрабатывает огромные массивы данных и настраивает внутренние параметры так, чтобы точнее предсказывать продолжение. Она не хранит внутри аккуратную библиотеку страниц, откуда по запросу достаёт нужный абзац. Усвоенные закономерности распределены по множеству параметров и внутренних представлений, поэтому знакомая форма не означает наличие конкретного проверенного источника.

Именно поэтому нейросеть умеет создавать новый текст, а не только находить старый. Она может объяснить понятие на другом уровне сложности, изменить структуру материала, сопоставить два подхода, выделить ошибки в инструкции. Она опирается на усвоенные закономерности языка и содержания. Но этот же механизм позволяет ей построить правдоподобную фразу там, где надёжного знания нет.

Возьмём короткое начало: «В соответствии с приказом Министерства просвещения Российской Федерации от...» В официальных документах после такой конструкции обычно стоят дата, номер и название. Языковая модель прекрасно знает форму. Если в контексте нет проверенного документа, она может продолжить её реквизитами, которые выглядят естественно. Для читателя это ссылка. Для модели — удачное продолжение последовательности. Разница обнаружится только при проверке.

Современные языковые системы не ограничиваются предварительным обучением на продолжении текста. После него модели дополнительно настраивают на выполнение инструкций, предпочтительное поведение, безопасность и работу с инструментами. Поэтому диалог давно не похож на бесконечное автодополнение: система умеет соблюдать формат, учитывать запреты, исправлять ответ после замечания и в некоторых режимах обращаться к внешним данным. Но это не превращает её автоматически в источник истины. Пользователю нужен верный и пригодный результат, а способность хорошо продолжать задачу и следовать инструкции сама по себе такой верности не гарантирует.

Отсюда появляется странное сочетание силы и ненадёжности. Система может разобрать сложную инструкцию, выдержать авторский тон, увидеть противоречие в двух абзацах — и ошибиться в простом факте. Может верно описать принцип оценивания и придумать номер

нормативного акта. Может улучшить объяснение по физике, но незаметно заменить одно условие задачи другим, потому что новый вариант лучше укладывается в знакомый шаблон.

Контекстом называют текст и другие данные, доступные системе при подготовке текущего ответа. В конкретном сервисе это могут быть служебные инструкции, сообщения пользователя, предыдущие ответы, прикрепленные материалы и результаты подключённых инструментов. Иногда отдельно существуют сохранённые настройки или память. Всё это важно различать: текущий контекст не равен знаниям модели и не равен надёжной долговременной памяти.

Если учитель передал фрагмент программы, нейросеть может опираться на него в текущей работе. Если документ не передан и внешняя проверка не включена, система использует то, что уже есть в её знаниях и контексте, либо достраивает недостающее. Большое окно контекста эту проблему не отменяет. Современные модели способны принимать очень объёмные материалы, но ёмкость окна не означает, что каждая строка будет одинаково хорошо найдена, сопоставлена и использована. Чем больше массив, тем важнее структура источников и явное указание на то, что действительно нужно для задачи.

Для учителя вывод простой: организацию источников нельзя заменять одним огромным диалогом. Фраза «мы это уже обсуждали выше» не является рабочим механизмом контроля. Критические правила должны находиться там, где их легко увидеть, проверить и передать заново. Поэтому позднее у системы появится ядро — короткий набор постоянных ограничений, паспорта курса и класса, а данные текущей задачи будут передаваться отдельно.

Есть ещё одно слово, которое часто создаёт лишнюю мистику, — внимание. В трансформерной архитектуре этот механизм помогает модели учитывать связи между разными частями последовательности и по-разному использовать их при обработке текста. Это не человеческое внимание и не осознанное сосредоточение. Для учителя важнее практический результат: система умеет связывать удалённые фрагменты, поддерживать структуру текста и учитывать условия, заданные раньше, но делает это не идеально.

Если инструкция содержит восемнадцать требований, несколько исключений и два противоречащих документа, модель может выдать текст, который выглядит полным, хотя одно ключевое ограничение потерялось. Чем труднее человеку проверить результат, тем опаснее успокаивать себя мыслью, что система «точно всё прочитала».

Поэтому архитектура строится не вокруг длины запроса, а вокруг управляемых переходов. Сначала определяется задача. Затем передаются необходимые источники. Потом один модуль создаёт результат в заранее оговорённой форме. Отдельный этап проверяет конкретные риски. Большую работу разбивают не для удобства нейросети как таковой, а чтобы учитель видел, где именно возникла ошибка.

Полезно различать обучение модели и подстройку в текущем контексте. Когда пользователь даёт два примера нужного результата, нейросеть может продолжить работу по их образцу. Основные параметры модели при этом не переписываются: примеры становятся частью текущего контекста и направляют следующий ответ. В новом пустом диалоге такая подстройка исчезнет, если сервис отдельно не сохраняет инструкции или память.

Отсюда растут многие разочарования. Учитель один раз долго исправляет стиль материалов, добивается ясного языка и решает, что нейросеть наконец научилась. Через неделю открывает новый разговор — и получает прежнюю гладкую пустоту. Машина не проявила упрямство. Просто рабочие правила остались в старом контексте.

Повторяемость появляется, когда удачные требования вынесены из переписки в отдельный документ. Вместо надежды, что система вспомнит предпочтение, появляется проверяемая строка в паспорте: «Инструкция ученику — одно действие на предложение; незнакомый термин объясняется до задания; скрытые требования запрещены». Такое правило можно прочитать самому, изменить, передать другой системе и проверить по результату.

Модель не обязана понимать, почему это важно для конкретного класса. Ей достаточно последовательно выполнять правило. Педагогический смысл остаётся у учителя.

В этом разделении нет попытки обесценить технологию. Оно как раз позволяет использовать её сильные стороны без лишней мистики. Языковая модель хорошо перерабатывает материал: сокращает, разворачивает, сопоставляет, меняет уровень объяснения, предлагает варианты, удерживает заданную структуру. Когда источники и критерии подготовлены, она снимает большой объём черновой работы.

Проблема начинается там, где способность создавать правдоподобное продолжение принимают за доступ к истине. У нейросети нет внутреннего красного огонька, который обязательно включается перед вымышленной датой. Уверенный синтаксис ей доступен всегда.

Следующее слово появляется. Потом ещё одно. Через несколько секунд ошибка уже оформлена как законченный абзац.

Глава 3. Ошибаться полным голосом

Человеческая речь выдаёт сомнение множеством мелочей. Пауза, оговорка, осторожное «кажется», взгляд на документ. Нейросеть умеет воспроизводить все эти сигналы, но их наличие или отсутствие не является надёжным показателем точности. Она может сказать «возможно» перед верным фактом и без колебаний назвать несуществующий источник.

Это свойство особенно опасно в профессиях, где форма сама несёт вес. Учитель привык различать черновик ученика, методическую рекомендацию, приказ, учебную статью. У каждого текста свой голос и свой уровень доверия. Нейросеть умеет имитировать почти любой из них. Ошибка, написанная языком официального документа, выглядит серьёзнее обычной ошибки. Она получает чужую форму и вместе с ней — незаслуженный авторитет.

Для таких случаев используют слова «галлюцинация» или «конфабуляция». Оба термина неточны, потому что переносят на машину человеческие состояния. В книге мы будем говорить проще: выдуманный или искажённый результат. Система создаёт сведения, которых нет в надёжном источнике, смешивает разные документы, приписывает автору чужую мысль, меняет условие задачи либо достраивает пробел наиболее правдоподобным вариантом.

Причина не сводится к небрежности разработчиков или плохому запросу. Сам механизм генерации тянет систему к продолжению. Современная настройка может научить модель останавливаться, попросить недостающие данные, отказаться от ответа или обратиться к инструменту, но это поведение не гарантировано в каждой ситуации. Если опоры не хватает, правдоподобное продолжение всё ещё может победить осторожность.

Ошибки бывают разными, и исправляются они тоже по-разному.

Самая заметная — чистая выдумка. Нейросеть называет книгу, статью, приказ или исследование, которых не существует. Иногда название складывается из реальных терминов и фамилий. Оно звучит настолько естественно, что человек ищет документ, меняет запрос, проверяет написание и лишь затем понимает: искать нечего.

Тоньше работает склейка. Система берёт номер одного документа, дату другого и содержание третьего. Каждая часть знакома, сочетание ложно. В школьной практике это опаснее откровенной фантазии: проверка по одному реквизиту может создать ощущение, что ссылка настоящая.

Есть ошибка подмены условия. Учитель просит переработать задание, сохранив проверяемое умение. Нейросеть упрощает текст — и заодно меняет само действие ученика. Вместо анализа появляется воспроизведение, вместо самостоятельного доказательства — выбор готового ответа. Материал стал доступнее, но учебная цель исчезла.

Есть ошибка согласия. Пользователь включает в вопрос неверное предположение, а система принимает его как часть задачи. Чем увереннее поставлен вопрос, тем легче не заметить подмену. Формулировка «почему новый порядок запрещает...» подталкивает к объяснению запрета, даже если такого запрета нет. Помощник, настроенный быть полезным и поддерживать разговор, иногда соглашается раньше, чем проверяет основание.

Наконец, есть ошибка уверенного обобщения. Нейросеть берёт частную практику, рекомендацию или решение одной организации и описывает его как общее правило. В образовании это происходит постоянно: локальное положение школы превращается в «требование законодательства», методический совет — в обязанность учителя, пилотный проект — в реформу для всей страны.

У всех этих случаев один неприятный признак: текст может быть качественным. В нём нет грубых грамматических сбоев. Логика абзаца выглядит последовательной. Иногда система сама приводит аргументы, подтверждающие выдуманное положение. Чем лучше написан ответ, тем больше дисциплины требуется от читателя.

Просьба «не выдумывай» полезна, но не создаёт гарантии. Фраза «используй только проверенные источники» тоже ничего не проверяет сама по себе. Если у системы нет переданных документов или надёжного доступа к внешним источникам, она работает с тем, что доступно в её знаниях и контексте. Если поиск, база документов или другие инструменты подключены, появляется новый слой проверки: что именно найдено, насколько источник надёжен, действует ли нужная редакция и не перепутаны ли проект, рекомендация и обязательная норма.

Поэтому в профессиональной системе проверяется не уверенность ответа, а его происхождение.

Для фактического утверждения должно быть понятно, на что оно опирается. Если речь идёт о законе или приказе, нужен первичный документ и действующая редакция. Если о содержании учебника — конкретный текст. Если о результатах класса — данные, которыми располагает учитель. Если система предложила собственную методическую конструкцию, её нельзя маскировать под официальную рекомендацию.

Так появляется дисциплина источников. Нейросеть должна отделять то, что извлечено из переданного материала, от собственного вывода и от предложения, созданного для обсуждения. В идеальном случае эти слои видны прямо в результате. Не потому, что читателю нравятся формальность, а потому, что разные слои требуют разной проверки.

Например, система готовит справку о новом порядке организации учебного процесса. Правильный маршрут начинается не с просьбы «расскажи, что изменилось». Сначала учитель или ответственный специалист собирает первичные документы. Затем отдельный модуль извлекает положения с указанием пунктов. Другой сопоставляет даты вступления в силу. После этого текст переводится на обычный язык. Финальная версия сверяется с оригиналом.

Если попросить ту же систему сначала подготовить справку, а потом просто «проверить себя» без источников и критериев, получится слабая проверка. Второй ответ легко наследует исходные допущения, формулировки и слепые зоны первого. Модель способна заметить часть ошибок, особенно когда ей точно сказано, что искать, но общий вопрос «всё ли здесь верно?» не создаёт независимой экспертизы.

Снизить эту зависимость можно организационно, даже если используется та же модель. Проверяющий контур получает исходные материалы, результат генерации и узкий список рисков. Ему не поручают «оценить качество вообще». Например, он должен перечислить фактические утверждения, найти для каждого основание в источнике и отметить неподтверждённые места. Это всё ещё не независимая экспертиза в строгом смысле, зато учитель получает конкретные точки для собственной проверки вместо общего вердикта «текст корректен».

Такой подход медленнее одного клика. Он всё равно быстрее, чем исправлять ошибку после того, как документ разослан родителям или задание попало детям.

В учебных материалах цена фактической ошибки понятна. Но есть ещё методическая уверенность, которую труднее заметить. Нейросеть без труда выдаёт объяснение, почему предложенное упражнение развивает критическое мышление, формирует метапредметные результаты или соответствует возрастным особенностям. Такие формулировки могут описывать намерение автора текста, а не реальное действие ученика.

Если школьник должен переписать готовые доводы в нужном порядке, само слово «анализ» в инструкции не превращает работу в анализ. Если весь ответ легко получить из первого абзаца текста, задание не проверяет понимание темы, сколько бы уровней сложности ни было указано в заголовке. Здесь нельзя ограничиться поиском неверных дат. Нужен модуль, который смотрит на действие ученика.

В дальнейшем мы научим систему задавать неприятные вопросы: что ребёнок делает сам, где может угадать, какой ответ считается доказательством понимания, не подсказывает ли формулировка решение, не заменено ли содержание красивой педагогической лексикой. Эти вопросы защищают от ошибки, которую не найдёт обычная проверка фактов.

Есть соблазн решить проблему полным запретом: не использовать нейросеть там, где нужна точность. Это не создаёт культуру работы с инструментом. Доступ к генеративным системам существует за пределами конкретного урока и конкретной инструкции, а гладкий ответ от запрета не становится менее убедительным. Полезнее научить человека видеть зоны риска и проверять их по-разному.

Нам нужен навык спокойного недоверия: не подозрительность ко всему тексту, а привычка различать цену ошибки. Объявление можно проверить одним способом, контрольную работу — другим. Подготовку характеристики ученика при помощи внешней системы вообще нельзя свести к качеству формулировок: здесь появляются персональные данные, полномочия взрослого и права ребёнка.

Уверенность нейросети не должна пугать. Она должна перестать нас убеждать.

На экране стояло точное название приказа, дата и номер. Ссылка открывала другой документ. Учитель удалил абзац целиком, а не попросил сделать его ещё убедительнее.

Глава 4. Почему хороший запрос не спасает плохую задачу

Первые книги и курсы о нейросетях приучили пользователей к простой мысли: качество ответа зависит от качества запроса. В ней есть правда. Точное задание обычно даёт лучший результат, чем случайная фраза. Указанный формат уменьшает хаос. Переданные критерии помогают системе не угадывать их самостоятельно.

Проблема возникает, когда из полезного правила делают универсальное объяснение. Нейросеть ошиблась — значит, пользователь плохо сформулировал запрос. Получился пустой урок — нужно добавить роль опытного методиста. Выдуман источник — следовало строже запретить выдумывать. Так ответственность незаметно переносится на человека, а устройство работы остаётся прежним.

Хороший запрос не исправляет отсутствие данных.

Если системе не передана действующая программа, подробное описание роли не сообщает ей нужную редакцию документа. Если неизвестно, что класс уже проходил, фраза «учитывай предыдущие знания» оставляет пробел. Если учитель сам не определил, какое действие должен выполнить ученик, нейросеть выберет привычную форму задания. Она может сделать это изящно. Цель от этого не появится.

Хороший запрос не устраняет противоречие.

Можно одновременно потребовать строго следовать источнику, сократить материал вдвое, сохранить все детали и написать простым языком для младшего возраста. Человек увидит, что условия конфликтуют, и начнёт выбирать. Нейросеть часто выполнит их частично, не сообщив, чем пожертвовала. Чем длиннее инструкция, тем легче принять внешнюю полноту за соблюдение каждого требования.

Хороший запрос не создаёт независимую проверку.

Если один и тот же диалог создал задание и тут же оценил его, проверка остаётся зависимой от того же контекста и тех же исходных допущений. Просьба «будь строгим экспертом» способна изменить тон ответа, но не происхождение данных. Для надёжности нужны отдельная функция проверки, конкретные критерии и исходный материал, с которым сравнивается результат.

Хороший запрос не даёт полномочий.

Нейросеть может написать убедительную характеристику ребёнка, предложить итоговую отметку или назвать работу списанной. Отточенная формулировка запроса не превращает такие решения в допустимые. Здесь ограничение проходит не по качеству генерации, а по праву человека передавать машине суждение, затрагивающее другого человека.

Наконец, хороший запрос не превращает плохую педагогическую задачу в хорошую. Если домашняя работа целиком выполняется генеративной системой, можно сколько угодно улучшать инструкцию к её созданию. Учитель получит безупречный материал, который не показывает самостоятельных знаний ученика. Технология лишь быстрее доведёт исходную ошибку до чистового вида.

Поэтому в этой книге промт займёт скромное место. Нам понадобится язык настройки: ясные инструкции, ограничения, описание входа и выхода. Но формулировка запроса останется деталью узла, а не самой системой.

Рабочая задача состоит как минимум из нескольких частей.

Сначала нужно понять, зачем выполняется работа и какой результат нужен. Затем определить входные данные: программу, тему, уровень класса, источник, ограничения. После этого выбрать исполнительный модуль и дать ему только необходимые полномочия. Он возвращает

результат в заранее понятной форме, затем включается проверка, а окончательное решение остаётся у человека. Сохраняется только то, что действительно пригодится снова.

Эта последовательность выглядит очевидной, пока не сравнишь её с обычным диалогом. В чате пользователь часто делает всё одним сообщением: просит определить цель, выбрать методику, написать материал, проверить его и оформить окончательно. Нейросеть отвечает одним полотном. Ошибка может возникнуть на первом шаге и пройти через все остальные, потому что этапы нигде не отделены.

Архитектура нужна прежде всего для видимости.

Если методический модуль предложил неверную цель, учитель исправляет цель до создания двадцати заданий. Если конструктор искажил источник, проверяющий указывает конкретное расхождение. Если адаптация упростила саму мыслительную операцию, материал возвращается на доработку. Становится понятно, где сломалась работа.

Это похоже на профессиональную деятельность самого учителя. Хороший урок редко рождается из одной длинной фразы в голове. Учитель сверяется с программой, оценивает место темы в курсе, выбирает способ объяснения, готовит задания, мысленно проходит урок, меняет то, что не сработает. Часть решений давно автоматизирована опытом и потому почти не замечается. Нейросети этот опыт не передаётся вместе со словами «ты опытный педагог».

Роль полезна только вместе с обязанностями. Фраза «работай как методист» слишком широка. Какой результат должен подготовить этот методист? На каких источниках? Что он не имеет права менять? По каким признакам видно, что работа закончена? Кому передаётся результат? Пока этих ответов нет, роль задаёт манеру речи, но не устройство работы.

Без обязанностей профессиональная роль превращается в костюм: система надевает нужный язык и продолжает угадывать.

Вместо коллекции эффектных запросов мы будем создавать паспорта модулей. Паспорт короткий. В нём есть назначение, входные данные, разрешённые действия, запреты, формат результата и проверка. Сам текст инструкции можно менять. Логика модуля сохраняется.

Допустим, нужен конструктор учебных заданий. Его назначение — создавать задания на уже определённое учебное действие. На вход он получает тему, цель, источник, возраст, уровень сложности и ограничения. Ему запрещено добавлять факты без основания и менять проверяемое действие. На выходе — задания, ответы, объяснение связи каждого задания с целью. После него работает проверяющий, который решает задания самостоятельно и ищет неоднозначность.

Здесь нет магической фразы. Есть устройство.

Такой модуль можно реализовать в простом чате. Паспорт хранится в текстовом файле. Учитель открывает отдельный диалог, вставляет правила и материалы, получает результат, переносит его в диалог проверки. Если сервис умеет сохранять постоянные инструкции, часть действий упростится. Если завтра сервис закроется, паспорт останется.

Архитектура защищает и от другой зависимости — от автора книги. Читатель не должен всю жизнь копировать чужие формулировки. Он должен понимать, почему модуль содержит именно эти поля, и уметь изменить их под свою работу. У преподавателя литературы и учителя химии будут разные источники, типы ошибок и способы проверки. Ядро системы общее, профессиональное содержание остаётся предметным.

Для этого придётся проделать работу, от которой многие руководства уклоняются: разобрать деятельность учителя на процессы. Не перечислить должностные обязанности, а увидеть внутреннее устройство. Где появляется цель? Когда материал становится заданием? Чем проверка знания отличается от проверки аккуратности? Какие сведения о классе действительно нужны, а какие нельзя передавать внешней системе? Как зафиксировать удачное решение, не превращая память в свалку старых переписок?

Ответы не помещаются в один запрос. И хорошо, что не помещаются.

Попытка сделать универсальный «суперпромпт учителя» обычно заканчивается многословной инструкцией, которую трудно проверить и ещё труднее поддерживать. В неё складывают всё сразу: педагогическую роль, стандарты, стиль, безопасность, формат, возраст, контроль, запреты. Через несколько страниц требования начинают конкурировать друг с другом. При изменении одного правила приходится искать его следы по всему тексту.

Модульная система устроена иначе. Постоянные правила лежат в ядре. Сведения о курсе — в паспорте курса. Данные о текущей задаче передаются отдельно. Проверяющий получает собственную инструкцию. Обновилась нормативная база — меняется нормативный слой, а не вся машина. Появился новый класс — создаётся новый паспорт, но методический модуль остаётся прежним.

Это и есть архитектура в рабочем смысле: решение о том, из каких частей состоит система, что знает каждая часть и как между ними движется задача.

Нам не потребуется рисовать сложные технические схемы. Маршрут можно записать одной строкой:

тема → учебная цель → действие ученика → материал → задание → проверка → решение учителя.

Если какой-то узел пропущен, система должна это показать. Если материал не опирается на источник, он не проходит дальше. Если задание не проверяет заявленное действие, его возвращают. Если результат затрагивает права ребёнка, машина останавливается до решения человека.

Такой порядок не делает работу безошибочной. Он делает ошибку обнаружимой.

В следующей главе мы на время отложим нейросеть и разберём саму работу учителя. Это может показаться странным: книга об искусственном интеллекте ненадолго перестанет говорить об искусственном интеллекте. Иначе машина снова окажется в центре, а педагогика превратится в набор задач для автоматизации.

На столе лежит чистый лист. Сверху учитель пишет не запрос, а два слова: «Что проверяю?» Пока ответа нет, чат можно не открывать.

Глава 5. Учебная цель появляется раньше материала

После разговора о языковой модели легко сделать следующий шаг слишком быстро — открыть нейросеть и начать собирать систему. Но первая деталь будущей машины находится не в чате и не в файле с инструкциями. Она появляется раньше, когда учитель решает, какое изменение должно произойти в работе ученика.

В школе это решение легко спутать с темой.

«Причастие». «Давление твёрдых тел». «Россия в эпоху дворцовых переворотов». «Обыкновенные дроби». Такие формулировки нужны: они удерживают место урока в программе, связывают его с учебником, календарным планированием и последовательностью курса. Но тема отвечает на вопрос, о чём пойдёт речь. Она ещё не говорит, чему ученик должен научиться.

Разница обнаруживается после звонка. Учитель рассказал материал, класс выполнил несколько привычных упражнений, в тетрадях появились записи. Если спросить, состоялся ли урок, внешние признаки будут на месте. Остаётся неудобный вопрос: что теперь умеет ученик из того, чего не умел сорок пять минут назад?

Ответ «знает тему» почти ничего не проясняет. Знать можно по-разному. Узнать термин среди четырёх вариантов. Воспроизвести правило. Объяснить связь своими словами. Применить способ в стандартной задаче. Перенести его в непривычную ситуацию. Найти ошибку в чужом рассуждении. Выбрать подход и обосновать выбор. Это разные результаты, хотя учебник открыт на одной странице.

Нейросеть умеет скрывать эту разницу особенно убедительно. Достаточно названия темы — и через несколько секунд появляются цель, задачи, ход урока и комплект заданий. Формулировки звучат методически привычно: «сформировать представление», «развивать умение анализировать», «способствовать воспитанию». Все поля заполнены, а учебная цель по-прежнему может оставаться невыбранной.

Проблема не в отдельных словах. Иногда «сформировать представление» точно описывает первый шаг знакомства со сложным явлением. Иногда урок действительно нужен для расширения словаря, постановки вопроса или подготовки к дальнейшему изучению. Ошибка начинается там, где широкая формула скрывает отсутствие решения. Учитель ещё не выбрал, какое действие станет результатом урока, а система уже производит материал.

В российской школе программа связывает планируемые результаты, содержание и условия обучения, а федеральные рабочие программы задают последовательность предметного содержания и результатов по годам обучения. Из этого нельзя механически выписать цель конкретного урока. Документы задают рамку. Учителю всё равно приходится определить ближайший шаг данного класса: что должно стать доступно именно сейчас и по какому действию это будет видно.

Здесь полезно отказаться от красивой формулировки и сначала записать цель грубо.

После урока ученик сможет различить два явления, которые раньше смешивал.

Сможет применить правило без подсказки в задаче знакомого типа.

Сможет прочитать исторический источник и отделить сообщение автора от собственной интерпретации.

Сможет объяснить, почему выбранный способ решения подходит, а другой приводит к ошибке.

Такая запись ещё не предназначена для рабочей программы или отчёта. Она нужна самому учителю. В ней слышно действие, границу и предполагаемый результат. Позже её можно привести к принятой терминологии. Если начинать с официально звучащей оболочки, есть риск оформить пустоту раньше, чем появится смысл.

У цели есть масштаб. На уровне образования она описывает результат нескольких лет. В программе предмета — продвижение по годам и темам. В разделе курса — освоение группы понятий и способов работы. На отдельном уроке — ближайшее изменение, которое реально подготовить и увидеть за ограниченное время. Языковая модель смешивает эти уровни и может пообещать за один урок «сформировать научное мировоззрение», «развить критическое мышление» и «воспитать ответственное отношение». Такие результаты не возникают по команде и не проверяются одним заданием в конце занятия.

Это не делает их ненужными. Школа работает с результатами разного масштаба. Личностное развитие, метапредметные умения, предметное знание не живут в отдельных кабинетах. Но крупная цель требует цепочки ситуаций, повторения, опыта и времени. Если система приписывает её одному конспекту, она не усиливает педагогическую работу, а производит торжественную документацию.

Учителю важно видеть расстояние между горизонтом и шагом.

Горизонт может состоять в том, чтобы ребёнок научился самостоятельно работать с информацией. Шаг конкретного урока — сравнить два источника и назвать основание, по которому одному из них в данной задаче можно доверять больше. Горизонт — математическая грамотность. Шаг — распознать величины в условии, выбрать отношение между ними и проверить правдоподобие ответа. Горизонт — осмысленное чтение. Шаг — найти место, где меняется позиция героя, и подтвердить это деталью текста.

Нейросети нужно передавать шаг. Горизонт остаётся в паспорте курса и помогает проверять направление.

Если дать системе только широкую цель, промежуток она заполнит наиболее привычными элементами: объяснением, вопросами, работой в парах, рефлексией. Сами по себе эти формы нормальны. Проблема начинается, когда форма начинает управлять содержанием. Групповая работа появляется потому, что часто встречается в конспектах. Рефлексия сводится к выбору смайлика. Вопросы проверяют память о только что сказанном. Урок выглядит современно и почти не меняет способность ученика действовать.

Поэтому первый модуль будущей системы нельзя называть конструктором урока. Сначала нужен модуль постановки задачи. Его работа скромнее и труднее. Он не пишет сценарий. Он помогает учителю проверить, достаточно ли определена цель.

На входе у такого модуля находятся место темы в программе, уже освоенный материал, ближайший планируемый результат, особенности текущего этапа и ограничения урока. На выходе — несколько возможных формулировок действия ученика, различия между ними и вопросы, которые требуют решения учителя. Модуль не выбирает цель окончательно. Он показывает варианты и обнаруживает пустые места.

Например, тема позволяет проверить как минимум три разных результата. Ученик может воспроизвести правило, применить его в стандартном случае или объяснить границу его применимости. За один урок, особенно первый, не всегда разумно требовать всё сразу. Система должна не складывать результаты в длинный список, а помочь увидеть выбор и цену каждого варианта.

Здесь проявляется профессиональное знание, которое нельзя извлечь из названия темы. Учитель знает, где класс спотыкается, какое слово вызывает ложную ассоциацию, почему формально простое задание отнимает десять минут, кто научился угадывать ответ по форме вопроса. Эти сведения не обязаны становиться персональными характеристиками детей. Для постановки цели достаточно обезличенного описания общего затруднения: класс выполняет алгоритм по образцу, но не распознаёт ситуации, в которых он применим.

Такое описание меняет задачу. Нейросеть уже не должна создавать ещё десять упражнений по образцу. Ей поручается помочь построить переход от распознавания к выбору способа. Материал, объяснение и проверка будут другими.

У цели есть ещё одна неприятная сторона: она ограничивает. Пока цель размыта, в урок можно поместить всё красивое и полезное. Интересный факт, видеоролик, игру, дополнительный текст, творческое задание. Каждая находка по отдельности выглядит оправданной. Вместе они могут занять время, в котором ученик должен был попробовать новое действие, ошибиться и получить обратную связь.

Нейросеть усиливает соблазн. Она способна за минуту предложить столько материалов, сколько учитель не успеет использовать за четверть. Поэтому архитектура начинается не с расширения возможностей, а с фильтра. Любой элемент будущего урока должен отвечать на простой вопрос: что он делает с заявленной целью?

Он вводит необходимое понятие. Даёт образец действия. Создает попытку. Обнаруживает затруднение. Позволяет проверить понимание. Помогает исправить ошибку. Если связь не находится, материал может быть хорошим и всё же лишним для этого урока.

Это ограничение быстро становится понятным, если помнить о реальном времени. Урок не растягивается от качества генерации. В классе остаются те же минуты, переходы, вопросы, опоздания и непредвиденные остановки. Избыток подготовленного материала не повышает качество: учитель либо начинает бежать, либо бросает половину задуманного, либо заменяет действия учеников собственной демонстрацией.

Цель нужна и для отказа.

Хорошая система должна уметь остановиться: этот фрагмент не обслуживает текущую задачу; это упражнение проверяет другое умение; эта часть требует предварительного знания; этот результат нельзя достоверно увидеть выбранным способом. По умолчанию языковая модель стремится продолжить ответ и предложить что-нибудь полезное. Поэтому условия остановки приходится задавать явно.

На первых этапах учитель будет делать это сам. Перед открытием чата он отвечает на четыре вопроса, пусть даже одной строкой: где тема находится в курсе, что уже умеют ученики, какое новое действие требуется и по какому результату станет видно продвижение. Не обязательно превращать ответы в бюрократический паспорт каждого урока. Их задача — удержать основание.

Позже часть проверки возьмёт методический модуль. Он сможет сопоставить цель с программой, заметить слишком широкий масштаб, предложить более наблюдаемое действие. Но даже идеально настроенная система не знает, какой шаг сейчас честен для данного класса. Это решение остаётся у учителя.

Цель начинает работать, когда по ней можно отсеять лишнее. Если после уточнения формулировки часть заранее придуманных упражнений становится ненужной, значит, основание урока стало точнее.

Глава 6. Что считается доказательством понимания

Цель становится рабочей, когда учитель понимает, по какому следу узнает результат. Не по настроению класса, тишине во время объяснения или количеству поднятых рук. Нужен ответ, действие или продукт, по которому можно судить: ученик продвинулся именно в том, ради чего строился урок.

Слово «доказательство» здесь не означает судебной строгости. Школьное обучение не даёт полной уверенности после одного задания. Ребёнок может угадать, растеряться, устать, неверно прочитать условие или справиться благодаря случайной подсказке. Речь идёт о разумном основании для педагогического вывода. Учитель видит не внутреннее понимание, а его проявление.

Это различие легко забыть, когда нейросеть создаёт оценочные материалы. Система получает тему и предлагает тест. Вопросы выглядят содержательными, ответы приложены, уровни сложности подписаны. Однако тест может проверять узнавание формулировки, тогда как целью было применение способа. Может требовать память о термине вместо понимания связи. Может содержать подсказку в самом вопросе. Может позволять исключить варианты по грамматике, длине или очевидной нелепости.

Материал оформлен как проверка, но не доказывает нужного результата.

Форма задания ничего не гарантирует. Тест способен проверять сложное различие, а развёрнутый ответ — воспроизведение заученного абзаца. Проект может показать самостоятельность, а может скрыть её под красивой презентацией. Устный ответ иногда раскрывает ход мысли, иногда лишь награждает уверенную речь. Учителю приходится смотреть глубже формата: какое действие должен совершить ученик, чтобы получить верный результат?

Если ответ можно дать, не выполняя заявленное действие, проверка сломана.

Допустим, нужно увидеть, умеет ли ученик устанавливать причинную связь в историческом материале. Вопрос «Назовите причины события» может проверить память о списке из учебника. Это допустимо, если задача урока состояла в воспроизведении основных положений. Для анализа причинности потребуется другое устройство: сопоставить обстоятельства, определить их роль, показать связь с последствием, отделить условие от непосредственного повода. Простое усложнение формулировки не поможет. Меняется сама работа ученика.

В математике можно получить правильный ответ, повторив знакомый алгоритм, и не понять, почему он подходит. В русском языке — расставить знаки по внешнему шаблону, не увидев синтаксической конструкции. В естественно-научном предмете — назвать результат опыта, потому что он прозвучал минуту назад, и не суметь предсказать изменение при другом условии. Ошибка проверки редко выглядит как грубая ошибка. Чаще задание честно измеряет что-то другое.

Поэтому после формулировки цели полезно дописать одну строку: «Я увижу результат, когда ученик...»

Не «получит пятёрку» и не «правильно ответит». Нужно назвать действие. Объяснит ход решения. Выберет признак и применит его к новому примеру. Найдёт противоречие между двумя сообщениями. Исправит ошибку и укажет её причину. Составит связное высказывание при заданных условиях. Предложит способ проверки собственного вывода.

Эта строка связывает урок и оценивание. Она не требует устраивать контрольную после каждого объяснения. Доказательством может быть короткий ответ, черновая запись, выбор с обоснованием, вопрос ученика, изменение первой попытки после обратной связи. Иногда учителю достаточно увидеть несколько работ и услышать ход рассуждения, чтобы понять, куда движется класс. Иногда нужен индивидуальный результат каждого. Это зависит от цели и цены ошибки.

Системе нужно передать, какой след считается значимым. Иначе она создаёт задания по наиболее распространённым шаблонам. В ответе появляются «вопросы на закрепление», но само слово «закрепление» ничего не говорит о проверяемом действии. Закреплять можно ошибочный способ, поверхностную формулировку и привычку угадывать ожидания учителя.

Здесь появляется важное различие между практикой и проверкой.

Во время практики ученик может получать опору: образец, памятку, наводящий вопрос, совместный разбор. Цель — освоить действие, поэтому помощь допустима и часто необходима. При проверке учитель убирает именно ту опору, без которой хочет увидеть самостоятельность. Если система смешивает эти режимы, она либо бросает ученика слишком рано, либо выдаёт за самостоятельный результат действие, выполненное по подсказке.

Одна и та же задача может работать по-разному. Сначала класс решает её вместе, проговаривая каждый шаг. Затем ученики решают похожую задачу с краткой памяткой. После — выбирают способ без подсказки. Внешне материал почти не изменился. Изменилась степень самостоятельности, а значит, и смысл результата.

Поэтому в описании задания недостаточно указать тему и уровень сложности. Нужны проверяемое действие, допустимые опоры, условия выполнения и признак успешности. Без этого маркировка «базовый уровень» или «повышенный уровень» остаётся этикеткой.

Сложность вообще часто понимают слишком грубо. Нейросеть увеличивает объём текста, добавляет числа, редкие слова или ещё одно условие и объявляет задание сложным. Ученическая работа может стать тяжелее, но не глубже. Иногда длинное условие проверяет устойчивость внимания, а не понимание предмета. Иногда необычный контекст мешает распознать уже освоенный способ. Иногда «творческое» задание требует ресурсов семьи и оформления больше, чем знания.

Учителю важно различать трудность и мыслительную нагрузку. Трудность может возникнуть из-за языка, объёма, непривычного формата, дефицита времени, моторных требований. Мыслительная нагрузка связана с тем, какое преобразование должен выполнить ученик: вспомнить, распознать, применить, сопоставить, обосновать, перенести, оценить. Эти слои пересекаются, но не совпадают.

Нейросеть можно использовать как дополнительный проверяющий проход: дать ей цель, задание и предполагаемый ответ и попросить искать обходные пути. Можно ли угадать ответ? Достаточно ли общего жизненного знания? Выдаёт ли один вариант себя длинной? Нужен ли материал урока? Совпадает ли способ получения ответа с заявленным действием?

Пока этого достаточно. Позже у системы появится отдельный «ученик-испытатель» с несколькими режимами проверки. Здесь важен сам принцип: он не изображает конкретного ребёнка и не строит психологический портрет, а испытывает конструкцию задания. Автор материала знает правильный ответ и невольно достраивает недостающий смысл; внешний проверяющий помогает этот зазор увидеть.

Такой проход тоже не выносит окончательный приговор. Модель может пропустить предметную тонкость, придумать несуществующий обход или сама решить задачу неверно. Её польза в дополнительном сопротивлении: материал проходит ещё одну проверку до того, как попадёт в класс.

Решающий вопрос остаётся человеческим: совпадает ли заявленная цель с тем действием, которое действительно выполняет ученик?

Если цель — научиться обосновывать, результат без обоснования не засчитывает её достижение, даже когда ответ верен. Если цель — выбрать способ, заранее указанный алгоритм снимает проверяемое действие. Если цель — работать с источником, вопрос, на который можно ответить по общей памяти, обходит источник. Если цель — увидеть границу правила, набор только стандартных случаев ничего не показывает.

Есть ещё одна ловушка генерации: система способна сразу выдать полный комплект — задания, ответы, критерии, шкалу. Из-за завершённой формы возникает ощущение, что оценивание уже спроектировано. Между тем критерий может описывать лишь внешний вид работы: полноту, логичность, грамотность, аккуратность. Эти качества имеют значение, но предметный результат они не заменяют.

Фраза «ответ полный и правильный» не объясняет, что должно присутствовать в ответе. «Приведены аргументы» не сообщает, какой аргумент относится к задаче и чем он подтверждается. «Решение выполнено верно» скрывает ключевые шаги, по которым можно различить понимание и случайный результат. Хороший критерий помогает двум людям увидеть в работе одно и то же основание для оценки.

Для учителя критерий служит ещё до проверки. Он показывает, возможно ли вообще наблюдать заявленную цель. Если невозможно описать признаки приемлемого результата, вероятно, цель пока слишком широка или задание с ней не связано.

Это не означает, что каждое учебное действие следует раскладывать на десятки пунктов. Избыточная детализация убивает целостность и создаёт новую бумажную работу. Иногда достаточно одного ясного основания: ученик выбирает формулу по физическому смыслу величин и объясняет выбор. Иногда нужны несколько признаков, потому что результат сложный. Архитектура должна помогать удерживать необходимое, не превращая педагогическое решение в автоматический подсчёт галочек.

Отдельно стоит работать с ошибкой.

Неверный ответ не всегда означает отсутствие знания. Ученик мог допустить вычислительную опisku после правильного выбора способа. Мог неверно понять одно слово. Мог смешать два близких понятия. Мог остановиться на полпути. Для обучения эти различия важнее общей метки «не справился». Они подсказывают, что исправлять.

Нейросеть способна сгруппировать обезличенные ошибки, сопоставить их с шагами решения, предложить вопросы для повторной попытки. Но сначала учитель должен решить, какие различия педагогически значимы. Иначе система создаст красивую классификацию, где живой ход мысли исчезнет под рубриками.

Доказательство понимания редко бывает одним числом. Даже отметка, выставленная по установленным правилам, сжимает сложный результат. Для текущей работы учителю нужен более подробный взгляд: что ученик сделал самостоятельно, где остановился, какую опору использовал, смог ли исправить ошибку после указания на противоречие.

Эта информация не обязана превращаться в огромный отчёт. Иногда она помещается в короткой помете на полях. Иногда — в следующем задании, выбранном точнее предыдущего. Нейросеть полезна, когда помогает увидеть структуру ответа и подготовить обратную связь. Опасной она становится там, где наблюдение незаметно становится готовым суждением о человеке.

Перед уроком учитель пересмотрел задания. Первые пять просили повторить правило разными словами. Шестое требовало выбрать способ и объяснить, почему он подходит. В листе осталось одно задание.

Глава 7. Где заканчивается подготовка и начинается решение

Большая часть учительской работы состоит из решений, которые со стороны выглядят как операции с текстом. Написать комментарий. Сформулировать характеристику. Выставить отметку. Подготовить сообщение родителям. Зафиксировать нарушение. Выбрать образовательный маршрут. Нейросеть умеет создавать текст, поэтому граница полномочий легко стирается: кажется, будто машина только помогает подобрать слова.

Иногда это действительно так. Учитель уже принял решение и просит сократить сообщение, убрать раздражение, сделать формулировку яснее. Содержание остаётся его. Система работает редактором.

Иногда решение незаметно рождается внутри генерации. Учитель передаёт несколько работ и просит определить уровень ученика. Нейросеть описывает «слабую мотивацию», «неустойчивое внимание», «недостаточное развитие аналитических навыков». Затем из этих формулировок собирается характеристика. На экране появился связный текст, но его основание никто не проверил. Машина перешла от обработки наблюдений к выводу о ребёнке.

Эта граница важнее качества запроса.

Можно составить очень точную инструкцию, потребовать осторожности, запретить медицинские диагнозы и попросить ссылаться только на данные. Всё равно остаётся вопрос: имеет ли система право делать вывод, который повлияет на отношение к ученику, оценку его возможностей или принятое школой решение? Если последствия выходят за пределы черновой подготовки, ответ нельзя прятать в формулировке промта.

Учитель не может опереться на одну специальную норму, которая перечисляла бы все допустимые и недопустимые способы применения генеративного ИИ в школе. На конец сентября 2026 года практические границы по-прежнему складываются из общих требований образовательного законодательства, локальных актов школы, прав обучающихся, обязанностей педагогического работника и правил обращения с данными. Новый инструмент не переносит ответственность за педагогическое решение на владельца сервиса.

Поэтому архитектуре нужен человеческий шлюз.

До него система может собирать, сравнивать, обезличивать, предлагать варианты и отмечать противоречия. После него находится решение, которое принимает уполномоченный человек. Чем серьёзнее последствия, тем заметнее должен быть переход.

Возьмём отметку. Нейросеть способна решить задание, сопоставить работу с критерием, указать пропущенные элементы и предложить предварительный результат. В некоторых типах строго формализованных упражнений это существенно ускоряет проверку. Но даже там остаются вопросы к исходному ответу, допустимым способам решения, техническим ошибкам, особенностям задания и установленному в школе порядку оценивания. Автоматическая цифра не становится отметкой только потому, что система посчитала баллы.

Учитель может использовать расчёт как материал. Он сверяет работу, видит расхождения, принимает решение и несёт за него ответственность. Если результат спорный, решение нельзя объяснить фразой «так определила нейросеть». Для ученика и родителя это не основание.

С характеристикой граница ещё жёстче. Машина может помочь привести в порядок фактические наблюдения: какие задания выполнены, какая динамика заметна, какие виды помощи использовались. Она может убрать эмоциональные оценки и двусмысленные фразы. Но выводы о личности, мотивации, семейной среде, способностях и будущем ребёнка не должны появляться из статистического продолжения текста.

Особенно опасны формулировки, которые звучат профессионально: «избегает интеллектуального усилия», «имеет низкую учебную мотивацию», «склонен к манипулятивному поведению». Одна такая фраза делает отдельный эпизод устойчивым свойством человека. Если основания не записаны отдельно, гладкий текст начинает выглядеть как вывод, хотя под ним может не быть достаточных наблюдений.

То же относится к подозрению в списывании или использовании генеративной системы. Текст может резко отличаться от прежних работ, содержать необычно ровные формулировки или другие признаки, которые вызывают вопрос. Технический детектор способен дать дополнительный сигнал, но его результат зависит от языка, длины, редактирования и конкретного инструмента. Такая метка не является самостоятельным доказательством нарушения.

Система может помочь подготовить проверочную беседу, выделить места, которые ученик должен объяснить, сопоставить текст с заявленными источниками и найти несоответствия. Решение о нарушении принимается по установленной процедуре и по совокупности оснований, а не по проценту в отчёте детектора.

Есть решения, которые нельзя передавать машине даже в форме «предварительного мнения», если предварительность неизбежно влияет на человека. Предположение о диагнозе. Вывод о наличии особых образовательных потребностей. Рекомендация перевести ребёнка на иной маршрут. Оценка семейного неблагополучия. Прогноз способностей. Дисциплинарная мера. В этих областях текст может запустить цепочку последствий раньше, чем кто-то вспомнит, что это был всего лишь черновик.

Удобная формулировка «человек всегда проверяет» здесь недостаточна. Проверка готового вывода часто зависит от первого впечатления. Если нейросеть уже назвала ученика немотивированным, последующие факты начинают читаться через этот ярлык. Архитектура должна требовать проверки и прямо запрещать системе создавать некоторые типы выводов.

Запрет можно записать прямо: не интерпретировать поведение как черту личности; не выводите причины из ограниченных наблюдений; не предлагать диагнозы; не определять виновность; не выставлять окончательную отметку; не рекомендовать меры, затрагивающие права ребёнка. При получении такой задачи модуль останавливается и возвращает вопрос учителю.

Остановка — полноценный результат работы.

Это непривычно, потому что цифровой инструмент обычно оценивают по объёму произведённого текста. Отказ выглядит как недоработка. В профессиональной системе всё наоборот: если задача выходит за пределы полномочий или данных недостаточно, остановка полезнее ещё одного убедительного абзаца.

Чтобы провести границу, полезно смотреть не на форму задачи, а на её последствия.

Редактирование сообщения родителю может показаться технической операцией. Но если в исходном тексте содержится обвинение, угроза или разглашение лишней информации, нейросеть способна лишь сделать их вежливее. Смысл останется. Значит, до редактора нужен этап проверки основания: что произошло, что подтверждено, какую цель преследует сообщение, какие сведения допустимо сообщать.

Подготовка протокола может выглядеть безопасной. Если система начинает заполнять отсутствующие детали правдоподобными формулировками, документ перестаёт отражать событие. Составление индивидуального задания кажется методической помощью. Если оно создано на основе переданной внешнему сервису медицинской информации, возникает другой риск, уже не связанный с качеством задания.

Границу удобно проверять тремя вопросами: есть ли у системы полномочие на такой вывод, допустимы ли переданные данные и какие последствия возникнут, если результат принять.

Первый вопрос возвращает решение тому, кто вправе его принимать. Второй проверяет, можно ли вообще передавать материал системе в данном виде. Третий заставляет заранее увидеть цену ошибки. Общая надпись «использовать ответ критически» эти проверки не заменяет.

Поэтому внутри личной педагогической системы задачи разделяются на подготовительные и решающие.

Подготовительная операция обратима. Черновик можно удалить, формулировку заменить, классификацию ошибок пересмотреть. Она не вступает в силу сама. Её результат получает учитель и использует как материал для собственной работы.

Решение меняет положение ученика или официальную запись о нём. Оно влияет на оценивание, доступ, обязанность, отношение, дисциплинарное последствие, образовательный маршрут. Здесь должен появиться человек, который имеет полномочия, видит контекст и способен объяснить основание.

Между этими зонами лежит немало смешанных задач. Например, нейросеть готовит обратную связь по работе. Если обратная связь указывает на конкретную ошибку и предлагает повторный шаг, она остаётся педагогическим материалом. Если в ней появляется оценка способностей — граница нарушена. Если система предлагает баллы по ясным критериям, учитель может использовать расчёт. Если она сама решает, насколько серьёзна ошибка и какую отметку заслуживает ребёнок в спорной ситуации, она уже действует за пределами подготовки.

Поэтому модуль должен возвращать текст вместе с основанием: какие данные использованы, какие критерии применены, какие места требуют проверки и какие решения система сознательно не принимала. Такая прозрачность не делает модель объективной, зато мешает готовому абзацу скрыть путь к нему.

Учитель тоже выигрывает от этой дисциплины. Когда основание записано, легче заметить собственную поспешность. Иногда выясняется, что раздражающая работа выполнена по двусмысленному заданию. Что «невнимательность» возникла из-за неясной инструкции. Что падение результатов совпало с переходом к другому формату, который класс ещё не освоил. Нейросеть не обнаруживает истину о ребёнке. Правильно настроенная система помогает не перепутать наблюдение и толкование.

Данные и документы мы разберём отдельно. Пока достаточно главного принципа: машине можно передать операцию, но ответственность за решение остаётся у человека.

На экране стояло: «Ученик систематически проявляет низкую мотивацию». Учитель вернулся к исходным записям. Там было только два факта: две работы не сданы; на третьей понадобилось дополнительное объяснение. В характеристику вошли именно они.

Глава 8. Какую работу стоит отдавать машине

После всех ограничений может возникнуть закономерное раздражение. Если результат нужно проверять, данные обезличивать, цели определять самостоятельно, а решения оставлять человеку, где обещанная экономия? Не проще ли продолжать работать как раньше?

Иногда проще.

Профессиональная архитектура начинается с права не использовать нейросеть. Новый инструмент не обязан присутствовать в каждой задаче. Если учителю быстрее исправить два предложения самому, отдельный модуль редактирования лишь добавит действий. Если материал уже выстроен и проверен годами, полная генерация новой версии может создать риск без пользы. Если задача требует разговора с ребёнком, никакой текстовый посредник не заменит присутствия взрослого.

Система нужна для того, чтобы освобождать внимание там, где машина действительно справляется с ограниченной операцией, а её ошибку можно обнаружить до последствий. Доля автоматизации сама по себе ничего не доказывает.

Для отбора задач полезно смотреть на четыре свойства: повторяемость, ясность критерия, обратимость и доступность проверки. Они не образуют математической формулы. Это способ остановиться до того, как удобство решит за нас.

Повторяемая работа обычно хорошо поддаётся передаче. Учитель регулярно сокращает собственные тексты, создаёт несколько вариантов однотипного упражнения, приводит формулировки к единому виду, выделяет ошибки в обезличенном массиве ответов, перестраивает материал под другой объём. Здесь уже существует понятный образ результата. Нейросеть ускоряет преобразование.

Но повторяемость сама по себе ничего не гарантирует. Выставление отметок тоже повторяется. Написание характеристик повторяется. Решение остаётся чувствительным, даже когда выполняется каждую неделю. Поэтому второй вопрос — насколько ясно можно описать качество результата.

«Сделай лучше» — слабый критерий. «Сократи текст до заданного объёма, сохрани определения и последовательность действий, не добавляй новых фактов» — рабочий. «Подготовь хорошие вопросы» оставляет машине право решать, что считать хорошим. «Создай вопросы, каждый из которых требует применения указанного правила и не содержит ответа в формулировке» уже позволяет проверить результат.

Чёткий критерий уменьшает пространство, в котором система вынуждена угадывать профессиональный замысел.

Третье свойство — обратимость. Ошибку в черновике карточки можно исправить до урока. Неверную формулировку в опубликованной характеристике вернуть сложнее. Ошибочный вариант тренировочного задания неприятен, но обнаружим. Несправедливая отметка или обвинение затрагивают человека. Чем труднее отменить последствие, тем меньше работы следует передавать машине и тем сильнее должен быть контроль.

Наконец, проверка должна быть доступнее самой задачи. Если для проверки нейросетевого результата учителю приходится заново выполнять всю работу с нуля, экономия исчезает. Иногда это всё равно полезно: система предлагает черновые варианты, а человек выбирает направление. Но нельзя строить постоянный процесс, в котором контроль дороже исполнения.

Надёжнее всего начинать с преобразования уже имеющегося материала. Система может сократить, развернуть, изменить уровень языка, сгруппировать, сопоставить, привести к формату или предложить варианты. Риск ниже, когда исходный смысл задан учителем или проверенным источником.

Хуже обстоит дело с задачами, где требуется установить новый факт, вынести суждение о человеке или учесть скрытый контекст. Здесь система начинает достраивать недостающее. Её текст может быть полезной гипотезой, но не готовым результатом.

Эта разница помогает разделить работу на несколько зон без цветных ярлыков и громких названий.

Есть операции, которые можно передавать после обычной проверки. Например, изменить структуру собственного объяснения, подготовить черновые варианты заголовков, убрать повторы, составить упражнения по уже заданной модели, создать разные наборы числовых данных при неизменном способе решения. Учитель знает исходный материал и быстро замечает искажение.

Есть задачи, которым нужен отдельный проверяющий контур. Подготовка контрольных материалов, адаптация текста, создание ответов и критериев, работа с фактами, сопоставление с программой. Ошибка здесь способна попасть в обучение или оценивание, поэтому одного чтения по диагонали недостаточно.

Есть область, где нейросеть может лишь помочь собрать материал, но решение остаётся полностью человеческим: отметки, характеристики, индивидуальная обратная связь в сложной ситуации, вывод о нарушении, выбор мер поддержки.

И есть задачи, которые не следует передавать внешней системе в исходном виде: работа с лишними персональными данными, закрытыми документами, медицинскими сведениями, материалами ограниченного доступа, паролями и идентификаторами. Здесь сначала меняют данные или сам способ работы.

Такой отбор лучше выполнять до создания модулей. Иначе учитель тратит время на настройку процесса, который не нужен или изначально опасен.

Полезно начать не со списка мечтаний, а с недели собственной работы. Не вспоминать всё, что теоретически умеет нейросеть, а записать повторяющиеся операции: что отнимает время, где чаще всего возникает раздражение, какие материалы приходится переделывать, какие решения требуют концентрации. Затем отделить механическую часть от профессиональной.

Проверка письменных работ, например, редко является одной операцией. В ней есть чтение ответа, обнаружение ошибки, сопоставление с критерием, решение о баллах, формулировка обратной связи, фиксация результата. Часть можно ускорить, часть нельзя отдавать, часть зависит от типа задания. Фраза «автоматизировать проверку» слишком широка и потому опасна.

Подготовка урока тоже распадается. Найти место темы в программе. Определить ближайший результат. Выбрать способ объяснения. Подготовить материал. Создать задания. Проверить время. Предусмотреть затруднения. Решить, что убрать. Нейросеть может участвовать почти в каждом этапе, но в разной роли. Если поручить ей «подготовить урок», все различия исчезнут в одном ответе.

Архитектура возвращает различия.

Учитель может оставить за собой постановку цели и выбор педагогического хода, а системе поручить создать три варианта практики по точным условиям. Может передать проверяющему поиск двусмысленных формулировок. Может попросить редактора сократить инструкцию, не меняя действия ученика. Каждый модуль получает небольшую работу, которую легче проверить.

Малый размер операции — одно из главных условий надёжности. Чем больше решений объединено в одном запросе, тем труднее понять, где результат отклонился. Система, которая сразу анализирует программу, придумывает цель, строит урок, создаёт задания, выставляет уровни и пишет домашнюю работу, выглядит мощной. При первой серьёзной ошибке её приходится разбирать целиком.

Отсюда следует неприятное правило: в начале архитектура может работать медленнее обычного чата.

Учитель описывает модуль, готовит входные данные, переносит результат на проверку. Быстрая генерация становится управляемым процессом. Экономия появляется позже, когда правила уже созданы, источники организованы, а повторяющиеся операции не требуют нового изобретения. Тот, кто обещает немедленно освободить половину рабочего дня, продаёт ощущение, а не систему.

Кроме времени есть ещё один ресурс — внимание. Иногда нейросеть не сокращает количество минут, но снимает пустую нагрузку. Учителю проще редактировать несколько предложенных вариантов, чем начинать с белого листа. Проще увидеть список возможных неоднозначностей, чем держать всё в голове. Проще передать системе техническое преобразование и сохранить силы для разговора, объяснения или разбора ошибки.

Иногда именно это и есть выигрыш: минуты почти не сократились, зато внимание осталось на разговор, объяснение и разбор ошибки.

Есть и обратный эффект. Генерация делает производство материалов почти бесплатным, поэтому материалов становится больше. Дополнительные карточки, варианты, памятки, комментарии, отчёты. Всё это нужно прочитать, выбрать, проверить, хранить. Если архитектура не умеет отказываться, нейросеть увеличит нагрузку под видом помощи.

У каждого модуля должны быть не только правила создания результата, но и условия, при которых он ничего не создаёт.

Если существующий материал соответствует цели, новый не нужен. Если различие между вариантами косметическое, достаточно одного. Если учитель не сможет проверить факты, генерация останавливается. Если операция занимает меньше времени вручную, система не используется. Если результат не будет применён, он не производится ради полноты комплекта.

Эти правила защищают от цифрового захламления. Папка с сотнями сгенерированных файлов не является профессиональной базой. База начинается там, где материал имеет назначение, источник, дату, понятный статус и причину хранения.

На этапе отбора не требуется создавать сложный реестр. Достаточно взять одну регулярную задачу. Например, подготовку тренировочных упражнений. Учитель описывает, что в ней повторяется, какие решения уже приняты, где чаще возникают ошибки и как быстро проверить результат. Если операция подходит, под неё создаётся первый модуль. Затем система проверяется на реальной теме, а не на демонстрационном примере.

После нескольких циклов станет видно, приносит ли она пользу. Возможно, модуль сократит время. Возможно, улучшит качество. Возможно, окажется лишним. Архитектура допускает удаление. Система не обязана расти бесконечно.

Поэтому хорошая система не накапливает модули ради количества. Если модуль не даёт понятной пользы, его удаляют так же спокойно, как лишний файл.

Дальше мы соберём основание машины: ядро правил, паспорта курса и класса, источники и рабочие модули. Но перед сборкой стоит зафиксировать критерий выбора. Нейросети передаётся та часть работы, которую можно ясно ограничить, снабдить материалом, проверить до последствий и вернуть человеку для решения.

В папке появился первый файл. В нём была одна строка: «Машине отдаётся операция, ответственность остаётся у учителя». Остальные модули пока не существовали.

Глава 9. Ядро начинается с запрета

К концу восьмой главы в папке лежал один файл с одной строкой: машине отдаётся операция, ответственность остаётся у учителя. Для полноценной системы этого мало. Но строка уже делала важную работу — ограничивала полномочия до начала выполнения задачи.

Первым узлом личной педагогической системы должно стать ядро. Слово звучит технически, хотя речь идёт о довольно простом документе — наборе постоянных правил, которые действуют независимо от темы урока, класса и текущего поручения. Ядро сообщает нейросети, какую работу она выполняет, на что опирается, где обязана остановиться и что не имеет права решать.

Обычно начинают иначе. В начале диалога пишут: «Ты опытный методист с двадцатилетним стажем». Формула создаёт нужную интонацию, иногда улучшает структуру ответа, но не даёт системе ни опыта, ни полномочий, ни доступа к реальной ситуации. Она не знает программу, не видит класс, не несёт ответственности. Роль без правил лишь подсказывает языковой модели, на какой тип текста следует быть похожей.

Ядро отвечает на более полезный вопрос: как системе работать.

Разница быстро обнаруживается в спорной ситуации. Нейросеть получает просьбу подготовить контрольную по теме и замечает, что в материалах нет критериев оценивания. Воображаемый «опытный методист» обычно восполнит пробел сам: предложит привычную шкалу, распределит баллы, оформит ответ. Система с ядром должна сообщить, что критерии не заданы, перечислить, какие решения отсутствуют, и остановить ту часть работы, где начинается произвольное оценивание.

Остановка может выглядеть менее полезной, чем готовая контрольная. На деле это признак управляемости: система показывает нехватку данных вместо того, чтобы прикрыть её гладким текстом.

Хорошее ядро невелико. Если в него поместить программу по всем предметам, особенности каждого класса, любимые форматы заданий, выдержки из законов и инструкции на все случаи, постоянные правила утонут. Документ станет длинным, противоречивым и неудобным для обновления. Учитель перестанет понимать, почему система поступила именно так. Исправление одного пункта начнёт менять результаты в местах, которые с ним вроде бы не связаны.

В ядре остаётся только то, что должно действовать всегда.

Прежде всего — назначение. Система помогает готовить и проверять материалы, организовывать информацию, предлагать варианты и обнаруживать слабые места. Она не принимает решений об учениках и не заменяет профессиональное суждение учителя. Это описание задаёт направление для всех последующих правил. Когда текущая просьба выходит за пределы назначения, система должна показать границу и вернуть решение человеку.

Следующий слой — отношение к фактам. Нейросеть обязана различать сведения из предоставленных источников, собственное обобщение и предположение. Если подтверждения нет, наиболее вероятная версия не становится установленным фактом. Если документ упомянут, но отсутствует среди материалов, его содержание считается неизвестным. Сведения из паспортов тоже дают контекст, но не доказывают текущее состояние ученика или класса и не заменяют наблюдение учителя.

Для педагогической работы этого правила недостаточно без требования указывать опору. Система должна показывать, на какой материал она ссылается: название файла, раздел, фрагмент, иногда страницу или пункт, если они доступны и надёжно распознаны. Не ради академической красоты. Учителю нужно быстро вернуться к месту, где утверждение можно проверить.

Здесь возникает первая ловушка. Языковая модель способна оформить ссылку даже тогда, когда не видит документа. Название будет правдоподобным, номер пункта — точным

на вид. Поэтому ядро запрещает создавать недостающие реквизиты. Нет страницы — значит, страницы нет. Не удалось установить пункт — система приводит короткий фрагмент и сообщает, что точная привязка требует ручной проверки.

Отдельный блок ядра касается данных. Внешняя нейросеть получает только ту информацию, которая необходима для операции и допустима к передаче. Фамилии, контакты, медицинские сведения, данные семьи, фотографии, закрытые школьные документы и иные лишние сведения не должны попадать в рабочий контекст. Если задача может быть выполнена после обезличивания, система сначала предлагает безопасную форму. Если обезличивание уничтожает смысл задачи, работа переносится в другой, разрешённый контур или выполняется вручную.

Ядро не стоит превращать в юридическую декларацию. Фраза «соблюдай законодательство» почти ничего не задаёт: модель не знает автоматически, какая редакция документа действует, какие локальные правила приняты в школе и где проходит граница допустимого в конкретной ситуации. Нужны операционные правила. Например: «Перед обработкой проверь, есть ли сведения, позволяющие определить ребёнка. Перечисли их отдельно и остановись, пока они не удалены или не заменены нейтральным описанием».

То же относится к педагогическим решениям. Общая просьба «не навреди» ничего не меняет. Рабочее правило звучит точнее: не выставлять итоговую отметку, не делать выводов о способностях, мотивации, диагнозе и семейной ситуации; не обвинять в списывании; не предлагать дисциплинарное решение. Система может сопоставить ответ с заданным критерием, выделить наблюдаемые признаки, показать противоречия. Последний шаг остаётся у человека.

Ещё одна часть ядра — порядок действий при неопределённости. Обычный чат обычно стремится завершить ответ даже при нехватке данных: расширяет толкование запроса, выбирает привычный вариант и идёт дальше. Ядро задаёт другой порядок: назвать недостающие сведения; отделить критические пробелы от некритических; показать, что можно сделать без них; остановить операции, для которых нет основания.

Это не означает, что система должна отвечать отказом на каждую неполную просьбу. В работе почти никогда нет идеального набора данных. Учителю нужен разумный черновик и возможность продвинуться. Поэтому ядро различает допущение и выдумку.

Допущение открыто. Система пишет, что принимает урок продолжительностью сорок пять минут, поскольку другое время не указано, и предлагает заменить параметр. Выдумка скрыта. Система молча строит сценарий на сорок пять минут и оформляет его как единственно подходящий. В первом случае учитель контролирует условие. Во втором узнаёт о нём, когда план перестаёт помещаться в реальность.

Правила вывода тоже относятся к ядру. Любой значимый результат должен возвращаться не только в готовом виде. Система указывает использованные источники, допущения, места неопределённости, выполненные проверки и решения, оставленные учителю. Такой хвост не нужен после каждой мелкой редакторской операции. Но там, где материал влияет на обучение, оценивание или официальный текст, он становится частью качества.

Есть соблазн записать всё это одним огромным обращением и вставлять его перед каждым заданием. Технически так можно работать. На практике длинный текст быстро обрывается уточнениями. В начале сказано не добавлять фактов, ниже требуется «сделать материал полным», ещё ниже — «не задавать лишних вопросов». Модель получает противоречивые инструкции и может следовать им непоследовательно, а учитель не сразу замечает, где возник конфликт.

Ядро требует редакторской дисциплины. Каждое положение должно выполнять отдельную работу. Если два пункта говорят об одном, остаётся более точный. Если правила спорят, заранее задаётся приоритет.

В педагогической системе приоритет можно выстроить без сложной нумерации. Сначала безопасность и запрет на действия за пределами полномочий. Затем обязательные требования и проверенные источники. После них — конкретное решение учителя по текущей задаче. Ниже находятся требования к формату, тону и удобству. Красивое оформление никогда не должно побеждать отсутствие основания, а просьба «ответь без уточнений» — отменять остановку перед рискованным выводом.

Приоритет особенно нужен, когда учитель сам торопится. Он может написать: «Сразу подготовь характеристику, вопросов не задавай». Ядро не спорит с человеком о цели, но не позволяет текущей формулировке отменить постоянную границу. Система может предложить нейтральную структуру, выделить факты из обезличенных записей, показать, каких подтверждений не хватает. Готовое суждение о ребёнке она не производит.

Именно поэтому ядро хранится отдельно от текущих заданий. Его не стоит растворять в длинной переписке: поздние формулировки могут вступить с постоянными правилами в конфликт. Если сервис поддерживает постоянные инструкции, документ можно разместить там. Если нет, ядро лежит в обычном текстовом файле и передаётся в начале нового рабочего разговора. Название функции меняется от системы к системе; принцип остаётся прежним.

Файлу удобно дать простое имя, которое всегда будет первым в папке: «00. Ядро». Ноль здесь выполняет чисто практическую работу — документ не теряется среди конспектов и карточек.

Внутри файла стоит указать версию и дату изменения. Не потому, что учитель должен вести разработку как крупная компания. Без даты через несколько месяцев невозможно понять, какое правило действовало, когда появился спорный материал. Версия помогает проверить новую редакцию и при необходимости вернуться назад.

Обновлять ядро следует редко. Правило одного курса отправляется в паспорт курса, условие конкретного класса — в паспорт класса, требование одного модуля — в инструкцию этого модуля. Критерий простой: в ядре остаётся только то, что нельзя забывать ни в одной задаче.

Перед рабочим использованием его проверяют не на красивом запросе, а на попытках сломать границы.

Системе дают недостаточный источник и просят назвать точный факт. Передают текст с явными персональными сведениями. Требуют выставить отметку без критериев. Просят сделать вывод о мотивации по двум несданным работам. Указывают не задавать вопросов, хотя данных нет. Предлагают сослаться на документ, которого система не видит.

Правильный ответ в таких испытаниях может быть коротким. Важно, чтобы система заметила нарушение, не замаскировала его и предложила безопасный следующий шаг. Если она выполняет просьбу вопреки ядру, правило нужно уточнить. Иногда дело не в формулировке: даже современные модели с большим контекстом могут непоследовательно соблюдать длинный набор инструкций. Архитектура должна учитывать это ограничение, а не перекладывать его на пользователя.

Ни одно ядро не гарантирует безошибочность. Языковая модель остаётся вероятностной системой. Она может пропустить противоречие, неверно применить правило или уверенно описать проверку, которой фактически не было. Ядро снижает произвол и делает часть ошибок заметнее. Надёжность повышают разделение операций, независимая проверка и решение человека.

Есть ещё одна опасность: ядро может законсервировать плохую практику. Если учитель впишет в постоянные правила привычное, но методически слабое требование, система начнёт воспроизводить его быстрее и аккуратнее. Технология не проверяет профессиональную ценность установки только потому, что она названа правилом. Поэтому ядро тоже проходит педагогическую ревизию.

Особенно внимательно нужно смотреть на формулы вроде «всегда добавляй рефлексия», «обязательно создавай три уровня сложности», «каждый урок заканчивай тестом». Они делают частный приём автоматическим ритуалом. Постоянным должно быть требование проверить достижение цели, а способ проверки выбирается по задаче. Постоянным может быть запрет на лишние данные, но не одинаковый формат для всех классов.

К концу первой редакции ядро обычно занимает одну или две страницы. Если оно разрослось до небольшой методички, его пора разделять.

Хорошее ядро удерживает границы: что известно, что неизвестно, какие действия разрешены и где система обязана остановиться. Для этого ему не нужна роль «опытного педагога» — нужны проверяемые правила.

Глава 10. Паспорт учителя — не автобиография

Ядро удерживает общие границы, но ещё ничего не знает о человеке, который будет пользоваться системой. Без этого нейросеть возвращается к усреднённому учителю: выбирает распространённые формы, привычный тон, случайную глубину объяснения. Иногда попадает. Чаще производит материал, который приходится долго приводить к собственному способу работы.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.