

Как перестать бояться неизвестности и начать собирать решения

Вводная лекция курса «Основы инженерного творчества»

Прикладная математика и информатика

Направление «Искусственный интеллект и системная аналитика»

Что ты делаешь, когда не знаешь, что делать?

**«Что ты делаешь, когда
не знаешь, что делать?»**

Пауза

Не отвечай вслух. Просто зафиксируй ответ.

Честный разбор: четыре привычные стратегии

Что обычно происходит

- **Зависаешь в телефоне** — симуляция деятельности.
- **Ждёшь, что кто-то скажет** — перекладывание решения.
- **Делаешь «что-то», чтобы быть занятым** — движение без цели.
- **Сдаёшься** — выученная беспомощность.

Это не слабость

Так мозг экономит энергию и избегает неизвестности. Стратегия работает, пока всё известно заранее. В реальной работе — не работает.

Я покажу другую стратегию

- Она сложнее привычной.
- Она требует усилий.
- Но она работает даже тогда, когда ответов нет.

Через полтора часа

У тебя будет выбор: оставить как есть — или начать собирать решения самому.

Полисистемный вход: где мы находимся

Пять уровней реальности

- 1 **Ты** — студент, который учится действовать.
- 2 **Группа** — среда, где ты пробуешь и ошибаешься.
- 3 **Вуз** — организация, которая даёт рамку.
- 4 **Рынок труда** — место, где проверяют твои навыки.
- 5 **Цивилизация** — переход к экономике, насыщенной ИИ.

Ключевая мысль

Каждый уровень предъявляет свои требования. Если ты видишь только один — теряешь остальные.

Через какие линзы мы смотрим

Одной линзы недостаточно

Линза	Вопрос
Педагогика	Как передать навык, а не только знание?
Психология	Что мешает студенту начать действовать?
ТРИЗ	Какое противоречие мы снимаем?
Системная инженерия	Из каких компонентов состоит решение?
Философия	В чём смысл этого действия?

Полисистемность

Собрав несколько линз, ты видишь объёмную картину, а не плоскую.

Главное противоречие

Тезис

«Хочу действовать, но боюсь ошибиться».

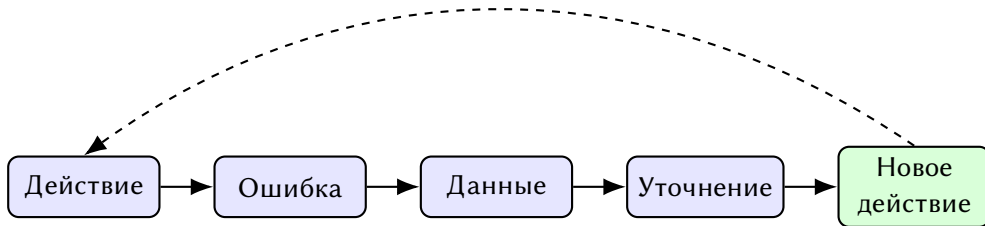
Или так

«Знаю теорию, но не знаю, как применить её на практике».

Это не слабость

Противоречие — это точка роста и место, где рождается новое решение.

Снятие противоречия



Не выбирать между «действовать» и «не ошибаться»

А сделать ошибку частью действия.

Результат

Ошибка перестаёт быть угрозой и становится топливом.

Метод, шаг 1: переформулировать задачу

Вместо «как это сделать?»

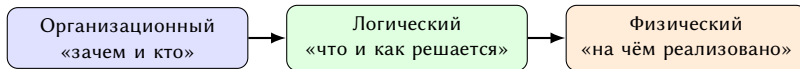
Спроси: «**зачем это нужно?**» или «**что должно получиться?**».

Пример

«Как сдать экзамен» → «Зачем мне эта дисциплина и что она даёт».

Инженерный факт

Большинство тупиков снимаются на этапе переформулировки задачи.



направление переформулировки — в сторону детализации →

Метод, шаг 2: найти ресурсы

Ресурс — это всё, что можно использовать

- Что я уже знаю и умею.
- Люди рядом: преподаватель, группа, эксперты.
- Что можно переиспользовать: старые решения, шаблоны, код.
- Интернет, LLM, время до дедлайна.

Почему ресурсы не видны

Ты ищешь готовое решение, а не собираешь его из того, что уже есть.

Метод, шаг 3: идеальный результат

Вопрос

«Как должно быть, чтобы задача решилась **сама**?»

Пример

Не «написать код», а «чтобы код писался сам» → тогда ты ищешь шаблоны, библиотеки, LLM и фреймворки.

Зачем это нужно

Идеальный результат даёт направление. Даже если не достигаешь его — ты идёшь к цели, а не в стену.

Искусственный интеллект — твой второй мозг

Для чего использовать LLM

- Проверка гипотез, а не копирование ответов.
- Генерация структуры и первых версий.
- Анализ собственных ошибок.
- Перевод сложного текста в понятную схему.

Правило

Используешь LLM как «ответчика» — ты её раб. Используешь как «редактора» — ты её хозяин.

Из чего собирается решение

Компонент	Вопрос
Цель	Зачем это нужно?
Задача	Что именно сделать?
Структура	Из каких частей состоит?
Ресурсы	Чем я располагаю?
Шаги	Что делаю и в каком порядке?
Проверка	Как пойму, что работает?

Любой проект — это архитектура

Видишь её — можешь собрать. Не видишь — собираешь кашу.

Чем junior отличается от middle

Junior

- Работает по инструкции и готовому ТЗ.
- Задача уже сформулирована кем-то.
- Нужен подробный план и контроль.
- Критерий: «сделал ли я то, что сказали?»

Middle

- Работает без инструкции.
- Сам формулирует задачу и видит структуру.
- Сам находит ресурсы и ограничения.
- Критерий: «решил ли я проблему?»

Разница — не в объёме знаний

А в способности действовать тогда, когда знаний нет.

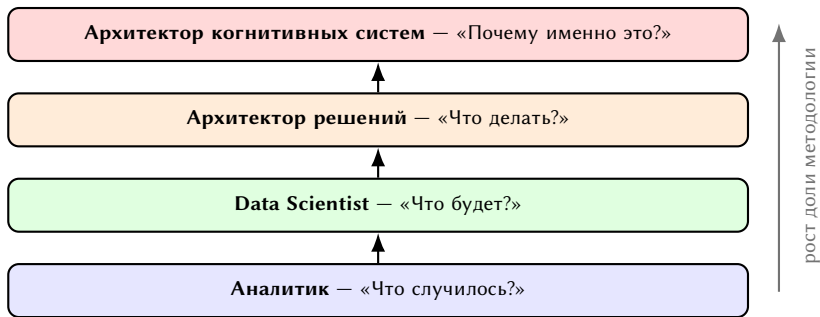
Что проверяют на разных уровнях

Уровень	Что проверяют	Ключевой навык
Junior	Умение выполнять по инструкции	Дисциплина и точность
Middle	Умение решить задачу без инструкции	Декомпозиция и метод
Senior / Архитектор	Умение менять правила и разрешать противоречия	Системное мышление

Задача курса

Тренировать переход: junior → middle → senior.

Эволюция роли в профессии



Как читать схему

Снизу вверх — от исполнителя к архитектору. Чем выше уровень, тем больше доля методологии и тем меньше рутинного исполнения.

Три уровня результата

Бронза

Сделать по инструкции.

Серебро

Сделать без инструкции: сам найти структуру и собрать решение.

Золото

Придумать, как сделать лучше, чем в инструкции.

Рынок

Бронзу постепенно забирает автоматизация. Ценят серебро и золото.

Кем ты станешь через два месяца

Портрет ИКР-студента (идеальный конечный результат)

- Не боишься задач без инструкции.
- Знаешь, с чего начать, даже когда не знаешь предмета.
- Видишь структуру и ресурсы там, где раньше был хаос.
- Собираешь решение не за месяц, а за сутки.
- Ты — архитектор своей роли, а не исполнитель чужих указаний.

Главное

ИКР — это не гений и не талант. Это навык, который можно собрать.

Твой школьный проект → метод

Шаблон пересборки

- 1 Вспомни свой школьный проект.
- 2 Переформулируй задачу: зачем он был нужен?
- 3 Найди три ресурса, которые ты не использовал.
- 4 Сформулируй идеальный результат.
- 5 Собери архитектуру: цель, структура, шаги, проверка.
- 6 Пропусти решение через LLM как через редактора.

Это шаблон

Не делай сейчас. Он остаётся у тебя — ты вернёшься к нему на практике.

Первое задание

Что сделать к следующему занятию

- 1 вспомнить один свой школьный проект.
- 2 Записать его суть в одном предложении.
- 3 Честно назвать самое слабое место.
- 4 Набросать, что должно было получиться.

Важно

Не пиши код. Только текст и схема. Мы будем пересобирать архитектуру, а не переписывать программу.

Что тебе мешает

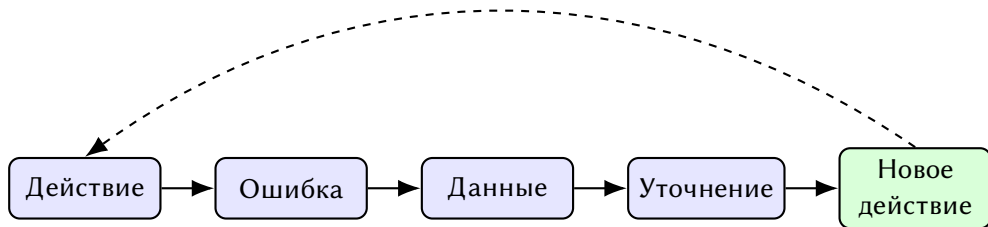
Три привычные фразы

- «Я не знаю, как начать».
- «Боюсь, что будет стыдно».
- «Я не успею».

Все три — не про задачу

Они про самооценку. Уберём страхи — задача решится.

Снятие страха ошибки



Тезис и антитезис

«Я боюсь ошибиться» → «Ошибка — это крах».

Снятие

«Ошибка — это данные для следующего шага».

Как мы будем работать

Этап	Формат
Лекции	Метод и разбор кейсов
Практика	Сборка решений своими руками
Проект	Защита перед экспертами
LLM	Помощник на всех этапах

Это не классический курс

Это сборка твоей способности собирать.

Ресурсы и экосистема

Что вокруг тебя

- Ты сам и твой опыт.
- Группа и преподаватель.
- LLM-ассистент.
- Эксперты и индустрия.

Что даю я

- Метод и инструменты.
- Шаблоны архитектуры.
- Чек-листы и обратную связь.
- Разбор твоих ошибок.

Главный инструмент

Это твоя голова. Я лишь помогаю её настроить.

Что я жду от тебя

Три простых правила

- 1 Задавай вопросы — даже те, что кажутся глупыми.
- 2 Пробуй — даже если страшно и не идеально.
- 3 Помогай другим, когда разобрался сам.

Не надо быть идеальным

Надо быть включённым. Остальное придёт.

Что ты уносишь с этой лекции

- Ты не боишься действовать в неизвестности.
- У тебя есть метод: переформулировать, найти ресурсы, собрать архитектуру.
- Ты видишь решение как систему, а не как хаос.
- LLM — твой помощник, а не хозяин.
- Ты понимаешь, чем junior отличается от middle и что делать дальше.

Сегодня мы не прошли ни одной темы

Мы настроили систему, на которой будут стоять все темы курса.

Мой пример

Пока я готовил этот курс, я применил тот же метод к своей реальной работе

- Два доклада по программе ТОП-ИТ:
 - **управленцам** — «Комплексная модель управления программой» (роли, ИКР, дашборд, противоречия);
 - **партнёрам на круглом столе** — «Выпускник, который не теряется».
- Задача звучала так: **рассказать о программе разным аудиториям.**

Дальше — как это делалось

Ниже — те же шаги метода, но на моей реальной задаче.

Мой пример: от задачи к ИКР

Шаг 1. Переформулировка

Не «как рассказать о программе», а **«кто такой студент в продуктивном подходе?»**.

Шаг 2. Ресурсы

Профподготовка Минцифры, проект ТОП-ИТ, мнения коллег.

Шаг 3. ИКР

«Студент, который не теряется». Доклад о нём — это и **ИКР**, и **ресурс**, проверенный на Минцифре.

Мой пример: шаги 4–6

Шаг 4. Противоречие

Программе нужны и **hard skills**, и **soft skills** — но кажется, что они **конкурируют** за время и внимание.

Шаг 5. Снятие (Aufhebung)

hard = soft skills: это **единая ткань**. «Код и диалог с заказчиком — одна деятельность».

Шаг 6. Вывод

Мы должны готовить **архитекторов**, а в базе — **системных и бизнес-аналитиков**.

Мой пример: результат

Сколько ушло

Одна неделя и один доклад.

Что это дало

Из этой работы и родилась идея **этой лекции.**

Вывод

Метод работает не в теории — он уже сработал на моей реальной задаче.

**Ты можешь быть студентом,
который сдаёт.**

**Или архитектором,
который собирает.**

Выбор за тобой

Возможность есть у каждого. Вопрос только в том, воспользуешься ли ты ею.