

# Советы по Upskilling

---

Автор: Justin Skycak

12 января 2025

*Советы о consistency, skills, дисциплине, the grind, пути, команде, миссии, мотивации, обучении и экспертизе.*

*Русский перевод · Оригинал: [justinmath.com](https://justinmath.com)*

# Советы по Upskilling

---

**Автор: Justin Skycak (@justinskycak) • 12 января 2025**

Советы о consistency, skills, дисциплине, the grind, пути, команде, миссии, мотивации, обучении и экспертизе.

---

## СОДЕРЖАНИЕ

**ВВЕДЕНИЕ** • У тебя есть потенциал

**CONSISTENCY** • Ты не ленив — у тебя просто нет привычки • Нет passion? Создай её. • Сделай привычку легко повторяемой • Самое трудное — просто начать • Как преодолеть боль • Если не получается тренироваться регулярно — делай это сразу после пробуждения • Тренировки должны быть короткими и частыми, а не долгими и редкими • Чуть больше consistency x чуть больше времени = огромный рост объёма и прогресса • Не реагируй остро на плохие дни • Ищи virtuous cycles • Защищай привычку

**SKILLS** • Hardcore skills — всегда ответ • Hack «Alien-Level Skills» • Важность prerequisites • Большинство людей не понимают, что могут систематически подниматься по skill trees • Самая неэффективная стратегия обучения • Укрепи свои чёртовы основы • Твои недостающие foundations ждут тебя • Потребление полезно лишь настолько, насколько оно помогает тебе производить • Активное делание — ключ к alpha • Всё имеет значение • Хочешь захватывающих возможностей? Учи математику и coding. • Если ты не одновременно technical и domain expert — ты underpowered • Domain expertise, математика, coding, коммуникация • Какую математику учить для skill stacking • Failure modes у людей, которые развивают math skills, но не капитализируют их через coding • Всегда найдётся плохая отговорка, чтобы не upskill'иться

**ДИСЦИПЛИНА** • Зачем тренироваться? • Магия — в том full-assed effort, которого ты избегаешь • Не откладывай тренировки из-за того, что цели кажутся слишком далёкими • В какой-то момент делать трудное становится легче, чем делать трудное — лёгким • Как культивировать дисциплину • Держи руки на камне

**THE GRIND** • Upskilling сложен — и это хорошо • Самая superior форма тренировки (Deliberate Practice) • Просто делай чёртову работу • Outsized success требует outsized работы • Трансформация некомфортна • Enjoyment — оптимизация второго порядка • Ability строится, а не «разблокируется» • Как ощущается max-efficiency тренировка • Необходимость прорабатывать конкретные примеры перед переходом к абстракции • Будь готов делать монотонную работу • Не недооценивай повышение интенсивности grind'a, но и не переоценивай последний виток • Когда больше объёма = больше прогресса • Провал — НЕ ключ к успеху • Не бросай себя в глубину • Меньше думай о чувствах — больше измеряй прогресс • Проблема со слишком сложными задачами • Дело не только в попытках — важно также достигать результата • В конце это всегда становится битвой воли • Цикл, в котором стоит находиться

**ПУТЬ** • Не зацикливайся на юношеских соревнованиях • Слишком легко переоценить своё значение, думая о percentile • 3 стадии развития таланта • В развитии таланта нет shortcuts • Эффект дебатов без базовых знаний • Потеря интереса иногда связана с отсутствием foundations • Если делаешь глупые ошибки — нужно больше практики • No train, no gain • Почему стоит push'ить себя • Держи ногу на газу • Ты — автомобиль • Что

делать, когда упираешься в ceiling • Учись любить игры, для которых ты создан • Compound усердный труд и удачу • Поставь себя в позицию, где можешь рисковать • Привяжи свой комфорт к реальной, осязаемой ценности • Если не можешь найти работу, которая тебя по-настоящему захватывает... • Как распределять bandwidth, пока ищешь свою миссию • Повторение ведёт к экспертизе, экспертиза ведёт к разнообразию

**КОМАНДА** • Если ты просишь кого-то быть твоим ментором — ты делаешь это неправильно • Создай давление на своего начальника, чтобы он придумывал тебе больше работы • Попади в правильную команду • Конкуренция как форма сотрудничества • Твоя цель — НЕ доказать, что ты умный, а решить проблемы • Произведи первое впечатление вкладом, а не критикой • Никогда не приходи с пустыми руками • Тебе нужен берсерк во главе

**МИССИЯ** • Выбрать хорошую задачу для работы • Чтобы понять, что тебя fulfills — нужно работать очень чертовски упорно • Формула «Progress Equals Pressure» • Люби то, что делаешь • Будь builder, а не только fighter • Строй там, где строительство создаёт больше возможностей для строительства

**МОТИВАЦИЯ** • Состояние «Blah»: почему ты в нём и как выбраться • Безразличие — часто просто overwhelm • Часто можно просто заставить себя полюбить вещи • Почему extrinsic motivation важна • Как стать super-producer • Как top performers поддерживают огромную нагрузку • Преодоление парадокса серьёзных тренировок • Радость от upskilling — как радость от физических упражнений • Ложь о том, что обучение должно быть приятным • Как изнурительная работа становится fun • Зачем стараться, если другие могут сделать лучше?

**ОБУЧЕНИЕ** • Величайший educational life hack: учиться заранее • Зачем быть намного впереди? Это покупает время. • В чём смысл instruction • Когда происходит обучение? • Не существует low-effort обучения • Величайший прорыв в науке об обучении за последний век • «Following along» против настоящего обучения • Необходимость практиковать retrieval из памяти • Порочный круг забывания • Одна из худших ошибок при учёбе • Сначала retrieval, потом рассуждение • Порочный круг context overload • Prereq Yo' Self Before You Wreck Yo' Self • Заполнение foundations — разница между ямами на дороге и непреодолимыми пропастями • Планируй путь top-down, но выполняй шаги bottom-up • Эффективный learning loop • Не раздувай feedback loop • Ловушки при обучении через проекты • Повторение должно ощущаться как вызов • Учись как поднимаешь штангу • Schooling против talent development • Обучение не обязательно должно быть синхронизированным, чтобы возникло товарищество • Sanity check для эффективных методов обучения

**ЭКСПЕРТИЗА** • Движущая сила экспертизы — долгосрочная память • Обучение — это память • Обучение — это строительство мостов, а не прыжки • Всё сводится к domain knowledge • Твой мозг должен тебя удивлять • Превращай магическое в механическое

---

## ВВЕДЕНИЕ

Я верю, что у тебя есть потенциал — даже если ты сам так не думаешь. Но ты должен заработать право увидеть его. Ты должен запустить процесс skill-building и серьёзно, последовательно придерживаться его на протяжении месяцев. Если ты это сделаешь, ты начнёшь замечать свою трансформацию и поймёшь, что потенциал у тебя действительно

есть.

Не ленись. Не придумывай оправданий. Не трать время в поисках hacks, чтобы добиться успеха без skills. Просто делай чёртову работу. Работай умно (эффективность) и усердно (объём) — и пожинай плоды.

Пойми: эта книга может немного встряхнуть тебя. Моя цель — вызвать в тебе реакцию. Если ты думаешь, что у тебя нет потенциала или work ethic, чтобы его реализовать, — моя цель давить на тебя, пока ты не скажешь: «Да пошёл ты, я тебе докажу». И тогда я сделаю всё возможное, чтобы помочь тебе это доказать.

Вот сделка. Вот какого коуча ты выбираешь с этой книгой. Supportive hard-ass. Не supportive pushover. Не unsupportive hard-ass. Supportive hard-ass, который хочет видеть, как ты делаешь трудную работу, необходимую для того, чтобы найти глубокое fulfillment в жизни, и предоставляет всё возможное руководство и поддержку для этого — но не боится говорить жёсткую правду и указывать на контрпродуктивные действия.

То, в чём люди больше всего нуждаются и чего меньше всего имеют в жизни, — это supportive hard-ass. Именно эту нишу я стремлюсь заполнить своим письмом.

---

## CONSISTENCY

### **Ты не ленив — у тебя просто нет привычки**

Если тебе трудно придерживаться demanding форм практики, временно забудь об эффективности и просто выработай привычку с менее эффективной, но более приятной формой практики. Хотя конечная цель — тренироваться эффективно и получать максимальный performance gain из ограниченного тренировочного времени, поначалу это будет ощущаться как тяжёлая работа, и ты, возможно, не захочешь так усердно работать — не потому что ты по природе своей ленив, а потому что ещё не выработал привычку.

Ты хочешь в итоге дойти до точки, где улучшение performance — твой основной фокус, а fun — оптимизация второго порядка. Но в начале можно оптимизировать для fun, чтобы помочь выработать привычку.

Возьмём силовые тренировки. Если только начинаешь и не хочется поднимать тяжёлые веса — всё нормально. Ещё не нужно. Твоя цель №1 — просто войти в зал и что-то сделать. Через неделю баскетбола, возможно, захочется попробовать bodyweight-упражнения — потом лёгкий weightlifting, а ещё через неделю, возможно, будешь готов бросить серьёзный вес на гриф.

То же самое с математикой. Если нет желания решать много сложных задач — нормально. Начни с YouTube-видео и Wikipedia. На следующей неделе попробуй решать несколько math-задачек каждый день. Через неделю — лёгкая алгебра. К этому моменту ты уже в тренажёрном зале, слегка разогрелся и готов работать всерьёз.

Как только привычка выработана — делай всё, чтобы сохранить её. Если хочется взять выходной, сделай хотя бы 10 минут — что-то незначительное, что поддерживает привычку. Привычка — это психологический щит против всех негативных чувств, которые пытаются отвести тебя от тренировок.

Итого: ты не ленив — у тебя просто нет привычки. Начни с малого. (Но если ты это знаешь и всё равно отказываешься вырабатывать привычку... тогда да, ты ленив.)

### **Нет passion? Создай её.**

Будь дисциплинирован, выработай привычку, compound, compound, compound. Развивай с ней отношения, вкладывай дополнительное время, когда скучно, приходи к ней, когда нужен эмоциональный outlet. Доверяй, что она будет расти на тебя и проникать в твою идентичность по мере того, как дни превращаются в недели, месяцы, годы практики и развития серьёзной экспертизы.

Это как развивать близкие человеческие отношения. Возможно, вначале не будет сильной связи, но вы «достаточно» ладите поначалу, и со временем узнаете друг друга так хорошо, проходите через столько всего вместе, что становитесь неразлучны.

Не думай слишком много о долгосрочной перспективе. Человек, ставший твоим лучшим другом, — ты, наверное, не знал этого в день знакомства. Делай краткосрочный прогресс — и долгосрочное разберётся само.

### **Сделай привычку легко повторяемой**

Когда начинаешь новую рутину, не превращай её в такое «большое дело», что делаешь один день и боишься повторить на следующий. Знаешь, что бывает с людьми, которые начинают новогоднее похудение с 3 часов в спортзале ежедневно? Они приходят один раз и не возвращаются!

Не делай так. Вместо этого начни с такого объёма работы, что не будешь бояться повторить его на следующий день. Поначалу не важно, что ежедневный объём слишком мал для достижения долгосрочных целей в нужные сроки.

По мере выработки привычки и адаптации тело и разум привыкнут, и наращивать объём станет легче. А когда привычка станет частью идентичности — ты сам захочешь этот объём увеличивать.

### **Самое трудное — просто начать**

Большинство skills можно тренировать. Но серьёзные тренировки обычно не приятны — поэтому большинство не тренируется. Это не означает, что в тренировках мало пользы. Просто многие не готовы вкладывать нужный труд.

Но самое трудное — всегда просто начать. Если ты плохо пишешь — просто сядь и пиши 15 минут каждый день. Возможно, неприятно вытаскивать мысли из головного бульона. Но это не значит, что так будет всегда. С практикой будет легче, и ты будешь замечать области для улучшения. И ты станешь лучше.

Прокрастинация нагнетает страх; начать — заставляет его рассеяться.

### **Как преодолеть боль**

Легче мотивировать себя тренироваться, когда видишь результаты. Но настоящую разницу делает момент, когда другие начинают замечать твои результаты и просить совета. Слышишь «Ты такой сильный, какая у тебя тренировка?» достаточно раз — и в твою идентичность просачивается ощущение, что ты сильный человек.

Это относится ко всем skill-областям, а не только к физическим упражнениям. Можешь делать это с математикой, coding, музыкой — чем угодно. Но самая трудная часть — это начало, до того как запустишь virtuous cycle.

Способ преодолеть боль — разумный тренировочный план, которому реально придерживаться. В начале нужно фокусироваться на consistency, а не на объёме. Нужно выработать привычку.

### **Если не получается тренироваться регулярно — делай это сразу после пробуждения**

Несколько причин, почему утренние тренировки работают так хорошо:

Во-первых, они защищают от неожиданных событий, которые разрушают любой запланированный распорядок. Если серьёзно настроен тренироваться — нужно встроить в тренировочный план устойчивость к хаосу.

Во-вторых, нет времени даже рассмотреть пропуск тренировки. К тому времени, когда стряхиваешь утреннюю вялость и в полной мере осознаёшь, что делаешь, — тренировка уже начата.

В-третьих, есть чёткий trigger и вознаграждение: пробудиться = время тренироваться, а после тренировки — душ, еда, ощущение гордости за себя весь день.

### **Тренировки должны быть короткими и частыми**

Предположим, у тебя есть 3 часа в неделю. Лучше тренироваться 30 минут шесть дней в неделю, чем 90 минут дважды в неделю.

Причины: во-первых, вырабатываешь привычку. Во-вторых, работаешь с пиковой продуктивностью — легко поддерживать фокус в короткой 30-минутной сессии, тогда как в долгой 90-минутной продуктивность резко падает по мере нарастания усталости. В-третьих, минимизируешь забывание между сессиями.

Оговорка: каждая сессия должна быть достаточно долгой, чтобы context-switching cost был пропорционально незначительным.

### **Чуть больше consistency x чуть больше времени = огромный прогресс**

Знаешь, в чём разница между 20 минутами 3 дня в неделю и 30 минутами 4 дня в неделю? Прогресс вдвое быстрее: 60 против 120 минут каждую неделю. Увеличь до 40 минут 5 дней в неделю — почти ещё одно удвоение: 200 минут в неделю.

### **Не реагируй остро на плохие дни**

Даже принимая правильные решения, можно иметь плохие дни. Важно сохранять consistency и не позволять одному плохому дню нарушить хорошую общую тенденцию.

Помогает зумировать назад и смотреть на прогресс на длинном временном масштабе. Долгосрочная игра — единственная, которая имеет значение.

При этом на каждый плохой результат нужно спрашивать: есть ли что-то, что можно из него вынести? Иногда — изъясн в рассуждениях, иногда — в предположениях. Одноразовые ошибки не мешают долгосрочному прогрессу, но повторяющиеся — да.

### **Ищи virtuous cycles**

Вещи compound'ятся — это факт жизни. Улучшаешь один аспект жизни — он имеет carryover-эффекты, которые сами имеют дальнейшие carryover-эффекты, и так далее. Важно использовать эти feedback loops и ориентировать их в позитивном направлении — потому что если они не движут тебя в позитивном направлении, они движут тебя в негативном.

Нет стабильного равновесия. Тебя будут втягивать в self-perpetuating циклы хочешь ты этого или нет. Делай всё возможное, чтобы попасть в virtuous cycles, а не vicious.

### **Защищай привычку**

Если я что и узнал о формировании и поддержании привычки, так это то, что нужно абсолютно оставаться consistent — даже (и особенно) в те времена, когда нужно снизить общий объём.

Когда кто-то сходит с пути к улучшению навыков, почти всегда одна и та же история: в какой-то момент он полностью упал с вагона и так и не смог вернуться.

Когда приходит время снова войти в ритм, гораздо легче ускорить медленный вагон, на котором ты сидишь, чем запрыгнуть на вагон, с которого ты полностью упал.

---

## **SKILLS**

### **Hardcore skills — всегда ответ**

Hardcore skills — самый большой bottleneck для улучшения твоей жизни и жизни других. Не важно, где ты находишься на спектре от эгоистичного до альтруистичного. Hardcore skills — всегда ответ.

Так много людей хотят оказать большое влияние на мир и на свою жизнь. Но желания недостаточно. Нельзя сделать ничего большого без больших skills. Конечно, некоторым везёт — рождённые в правильной семье, в правильном месте, в правильное время. Но даже для этих людей разница между относительно большим impact и абсолютно большим impact («вмять вселенную») всё равно сводится к skill-building.

Развитие hardcore skills — один из величайших social mobility hacks. Даже если твоя семья не имеет связей, ты можешь компенсировать это, развивая реальные skills.

### **Hack «Alien-Level Skills»**

Сильные технические навыки могут быть огромным power-up'ом, который выделяет тебя. Ты можешь предоставлять ценность, которую никто другой не может, и получать за это recognition.

Одна из вещей, которая мешает людям использовать этот hack, — они не инвестируют в построение широких технических foundations. Например, распространённый аргумент против широких математических foundations: «Зачем учить математику сейчас, если можно подождать и backfill нужную математику по мере столкновения с конкретными задачами на работе?»

На первый взгляд это звучит разумно. Но на практике подход «подождать и backfill» снижает именно твои шансы воспользоваться alien-level skills hack'ом.

Если работаешь в math-adjacent поле без хорошей математической базы:

- Будешь недооценивать, как часто появляются mathy задачи.
- Даже если встретишь mathy задачу — не будет достаточно базы, чтобы определить нужный подход.
- Даже если знаешь подход — задача может быть передана кому-то с большей базой, и ты окажешься на проигрывающей стороне Matthew effect of accumulated advantage.

- Даже если передавать некому — возможно, не хватит времени на spin up и выполнение.

### **Важность prerequisites**

Иметь prerequisites на месте — это разница между тем, что кажется запутанным и недоступным, и «Подождите... это всё?»

Легко думать, что тебе не хватает learning ability, когда на самом деле не хватает prerequisite знаний. Различия в learning ability существуют, но их часто путают с наличием или отсутствием prerequisite знаний.

В целом, способ «открыть» вещи, которые кажутся недоступными, — укрепить свои prerequisite способности.

### **Большинство людей не понимают, что могут систематически подниматься по skill trees**

Они пробуют кучу случайных вещей в случайном порядке, непоследовательно, видят мало прогресса и быстро сдаются. Они думают, что не могут взобраться на дерево, хотя на самом деле просто пытаются достать высокую ветку прямо с земли.

Они не понимают, что у каждого skill tree есть ствол и нижние ветви, по которым нужно взобраться сначала. Они не понимают, что это дерево.

### **Самая неэффективная стратегия обучения**

Самая неэффективная стратегия — снова и снова спрашивать «почему я такой тупой». Более эффективная стратегия — спросить «какие (prerequisite) знания мне не хватает» и вернуться, чтобы их восполнить.

### **Твои недостающие foundations ждут тебя**

Если думаешь, что тебе не нужно подниматься по skill tree и осваивать prerequisites — не надо. Просто иди и делай [трудное дело]. Твоим недостающим foundations всё равно. Они будут терпеливо ждать тебя, пока ты не будешь готов их признать.

### **Потребление полезно лишь настолько, насколько помогает производить**

Если хочешь серьёзно научиться кататься на коньках, нужно получить коньки и поверхность, и делать упражнения, нарастающие по сложности. Нельзя научиться кататься, просто потребляя видео и читая об этом.

Это явление очевидно и обобщается на каждую skill-область. Если хочешь иметь возможность что-то производить — нужно практиковать производство, а не только потребление.

### **Активное делание — ключ к alpha**

Активное делание (в отличие от пассивного потребления) — ключ к эффективному обучению. Но это больше. Это также ключ к alpha — развитию edge.

Многие потребляют. Меньше активно делают. Ещё меньше пробуют сложное. И ещё меньше строят foundational skills, необходимые для успеха в этих сложных вещах.

Alpha также compound'ится. Когда успешно делаешь трудное дело, знания и ресурсы, которые ты приобретаешь, позволяют тебе преуспеть в ещё более трудных вещах. Твой edge становится острее — не тупее — с использованием.

## **Всё имеет значение**

Hard и soft skills, big-picture и detail-oriented мышление, technical и creative способности — ВСЕ они крайне ценны. Лучшие результаты приходят от того, что solid в каждой из этих категорий и world-class там, где ты естественно блестяешь.

Опирайся на свои strengths И укрепляй свои weaknesses. Ты максимизируешь производство, а не сумму. Всё имеет значение.

## **Хочешь захватывающих возможностей? Учи математику и coding.**

Изучение продвинутой математики и coding открывает карьерные двери, о существовании которых ты даже не подозреваешь. Все знают, что будущее уже здесь — просто очень неравномерно распределено. Знаешь, где оно сконцентрировано? У тех, кто безумно skilled.

Если у тебя серьёзные math chops, coding chops и глубокая domain expertise в другой дисциплине — ты compound'ишь три порядка величин. Будь one-in-a-hundred в каждом — и ты один на миллион в целом. Будь one-in-a-thousand в каждом — и ты один на миллиард.

**TLDR:** Преследуй область, которую любишь, но одновременно становись настолько безумно технически skilled в математике и coding, чтобы применять их к своей области интереса инновационным образом.

## **Если ты не одновременно technical и domain expert — ты underpowered**

Если ты domain expert, но не можешь сам строить — ты severely underpowered. Аналогично, если ты technical, но полагаешься на кого-то другого в domain expertise — тоже severely underpowered.

Лучше всего быть technical domain expert и работать с другими technical domain experts, когда это возможно.

## **Domain expertise, математика, coding, коммуникация**

- **Domain expertise** для идентификации важной проблемы и создания видения решения
- **Математика и coding** для его построения
- **Коммуникация** для доставки

Без domain expertise выберешь неважную или неразрешимую проблему. Без математики и coding ограничишься тем, что кто-то или что-то другое (с относительно небольшой domain expertise) может построить для тебя. Без коммуникационных навыков твоё решение не будет понято и принято.

## **Какую математику учить для skill stacking**

Большинство ценности приходит от:

- Абсолютной надёжности в applied math, которая постоянно встречается в инженерии: линейная алгебра, calculus-based probability и statistics, алгоритмы и т.д.
- Широкого знания продвинутых подходов к математическому моделированию: дифференциальные уравнения, machine learning и т.д.
- Знания основ доказательств.
- Глубокого погружения в математическое моделирование, поддерживающее конкретные области, которые тебя интересуют.

## **Failure modes у людей с math skills, которые не капитализируют их через coding**

Два trend'a у людей, которые skilled в математике, но не капитализировали на этом: 1) как правило, не очень skilled в coding, и 2) не хватало дисциплины работать над тем, что нужно сделать, но не захватывает интеллектуально.

Четыре failure mode'a даже у mathematical-inclined людей:

- Сложность с complexity, которая растёт так, что уже не умещается в голове.
- Нежелание принимать практические constraints задачи.
- Отвлечение на low-ROI features/детали.
- Нежелание делать «монотонную» работу.

## **Всегда найдётся плохая отговорка**

Сильные foundational skills всегда обеспечат огромное преимущество. Большинство людей просто не хотят вкладывать труд и ухватятся за любую отговорку.

Если сейчас отказываешься развивать серьёзные технические навыки из-за AI или чего-то ещё — не обманывай себя. Ты никогда и не собирался этого делать. Ты всегда собирался ухватиться за ближайшую отговорку, и AI — это отговорка, на которую ты наткнулся.

---

## **ДИСЦИПЛИНА**

### **Зачем тренироваться?**

Большинство skills можно тренировать. Но серьёзные тренировки обычно не приятны, поэтому большинство их избегает.

Если плохо пишешь — просто сядь и пиши 15 минут каждый день. Со временем станет легче, будешь замечать области для улучшения. И станешь лучше.

### **Магия — в том full-assed effort, которого ты избегаешь**

Поразительно, сколько людей рационализируют избегание skill development на основании «нет, чувак, я лучше буду жить». Дело в том, что можно делать серьёзный прогресс по практически любому skill tree, если просто вкладывать 30 минут сфокусированных усилий каждый день. Но это должен быть полностью сфокусированный full-assed effort — и нужно постоянно повышать уровень сложности по мере роста способностей.

Магия, которую ты ищешь, — в том full-assed effort, которого ты избегаешь.

### **Не откладывай тренировки из-за дальних целей**

Перестань искать tricks. Просто становись лучше. Это и есть trick. Hack — буквально просто upskill'иться как можно эффективнее. Нет skill — нет результатов. Нет тренировки — нет прогресса.

Да, upskilling требует много работы и поначалу может казаться недостижимым. Но как только реально сядешь в вагон и почувствуешь, как быстро движешься, можешь обнаружить, что цели ближе, чем думал.

Что МОЖНО сделать — работать эффективно и сжимать необходимый объём работы в удивительно короткое время.

## **В какой-то момент делать трудное становится легче, чем делать трудное лёгким**

Условие, чтобы заставить себя что-то делать: внутренняя willpower  $\geq$  внешнее friction. Если это условие не выполнено, сделать его выполненным можно, уменьшив friction и/или увеличив willpower.

Твоя жизнь — как большая codebase: если трудно внедрить новое поведение в какой-то области, refactor эту область, чтобы было легче строить.

Но cost-cutting имеет пределы. В какой-то момент делать трудное становится легче, чем делать трудное лёгким. Тогда нужно собрать willpower, чтобы преодолеть оставшееся friction.

## **Как культивировать дисциплину**

Когда есть что-то, что знаешь, что нужно делать, но не можешь заставить себя делать — какая-то привычка тянет тебя от продуктивного действия. Нужно сломать непродуктивную привычку и выстроить counter-habit, чья сила постепенно станет достаточной, чтобы полностью взять верх над исходной привычкой.

Ты прерываешь momentum на негативной привычке и создаёшь momentum к позитивной.

## **Держи руки на камне**

Люди будут делать невероятные mental gymnastics, чтобы убедить себя, что делать лёгкое, приятное дело, не связанное с их предполагаемой целью, как-то двигает needle больше, чем делать трудное, неприятное дело, прямо связанное с этой целью.

Если хочешь двигать needle по цели — нужно концентрировать усилия непосредственно на этой цели. Другого пути нет.

---

## **THE GRIND**

### **Upskilling сложен — и это хорошо**

Upskilling требует работы. Это feature, а не bug. Upskilling повышает agency, а outsized agency лучше держать в руках людей, которые знают, что значит вкладывать outsized работу.

### **Самая superior форма тренировки: Deliberate Practice**

В области talent development абсолютно нет дискуссий о самой superior форме тренировки. Это «deliberate practice»: mindful repetition на performance задачах прямо за пределами своих возможностей.

Deliberate practice — о том, чтобы делать performance-improving adjustments при каждом повторении. Отдельная корректировка мала и даёт небольшое улучшение — но когда compound'ишь эти небольшие изменения над огромным числом action-feedback-adjustment циклов, получаешь массивные изменения и массивные performance gains.

Почему тогда кто-то практикует другие формы? Ответ, возможно, в самых ударных двух предложениях всей talent development research: «Deliberate practice требует усилий и не является inherently приятной. Индивиды мотивированы практиковаться, потому что практика улучшает performance.» (Ericsson, Krampe & Tesch-Romer, 1993)

Deliberate practice — НЕ бездумное повторение. Если делаешь одно и то же снова и снова — делаешь deliberate practice неправильно.

### **Просто делай чёртову работу**

Если хочешь развить серьёзные skills — нужно выполнять интенсивные, изнурительные тренировки. Думаешь, ты слишком хорош для grunt work? Слишком умён, чтобы слушать feedback коуча? Тогда иди — докажи свою innate greatness, или провались достаточно раз, чтобы в голову вбились немного смирения.

Ты можешь либо тратить время, споря с коучем о тренировочном режиме, либо использовать это время, чтобы опустить голову и делать чёртову работу.

### **Outsized success требует outsized работы**

Хочешь сделать что-то, что выделяет тебя? Нужно будет работать усерднее, чем большинство. Другого пути нет.

Outsized success требует не только достаточно времени и усилий, но и продуктивной работы. Нужно работать усердно И умно.

### **Трансформация некомфортна**

Если не push'ишь себя за пределы зоны комфорта — не улучшаешь performance. Просто так. Strain может быть неприятен. Он изматывает и оставляет усталость. Можешь чувствовать себя слабым или даже тупым, тренируя интеллектуальный skill. Но это абсолютно необходимо.

Трансформация некомфортна. Но помни: дискомфорт необходим для улучшения performance — но дискомфорт сам по себе не всегда указывает, что ты занимаешься performance-improving активностями. Нужно также уметь преодолеть вызов.

### **Enjoyment — оптимизация второго порядка**

Если стремишься максимизировать learning per unit practice time — enjoyment является оптимизацией второго порядка, которая часто противоречит первостепенной: deliberate practice.

Fun — дополнение, а не замена deliberate practice. Ranking по эффективности: enjoyable deliberate practice > unpleasant deliberate practice >>> другие enjoyable формы тренировки > другие unpleasant формы тренировки.

### **Ability строится, а не «разблокируется»**

Один из наиболее вредных мифов в образовании — что ability нужно «разблокировать» любопытством и интересом, а не «строить» через deliberate practice.

Iron Man — гораздо лучшая mental model. Постепенно строй свой skillset. Не существует скрытой суперсилы, которая разблокируется за один день. Ты должен строить skillset.

### **Как ощущается max-efficiency тренировка**

Max-efficiency тренировка должна ощущаться достаточно scaffolded — что learning tasks сложны, но ещё достижимы в разумные сроки.

Когда развиваешь skills с пиковой эффективностью — максимизируешь сложность training tasks при условии, что в итоге успешно преодолеваешь эти трудности своевременно. Confidence становится скорее «ретроспективной» вещью.

## **Необходимость прорабатывать конкретные примеры перед абстракцией**

Многие learners не понимают, что grinding через конкретные примеры даёт интуицию, которую не получишь, перепрыгнув сразу к абстрактным идеям.

Если переходишь прямо к самым абстрактным идеям — ты как ребёнок, который читает книгу известных цитат о жизни и думает, что понимает всё о жизни через эти цитаты. Способ понять жизнь — накапливать жизненный опыт.

## **Будь готов делать монотонную работу**

Чтобы хорошо понять, что действительно важно, нужно охватить задачу — что обычно требует испачкать руки и делать достаточно manual grunt work для выработки сильной интуиции.

Многие люди оправдывают избегание grunt work тем, что он скучный и они уже всё продумали в голове. Они не понимают, что контур проблемного пространства в их голове не соответствует реальности.

## **Важно больше объёма = больше прогресса**

Больше объёма = больше прогресса — при условии, что работаешь продуктивно и не выгораешь. Если устал, голова идёт кругом и делаешь много глупых ошибок — время остановиться. Если настолько выдохся, что между вопросами неизбежно отвлекаешься — время остановиться.

Но до этих точек: делай больше, учи больше, быстрее прогрессируй к долгосрочным целям.

## **Провал — НЕ ключ к успеху**

Провал слишком переоценивается как ключ к успеху. Провал движет тебя к успеху лишь в той мере, в которой ты из него учишься. Никогда не следует проваливаться дважды по одной причине.

Всегда анализируй свои ошибки. Поддерживай гигантский locus of control и смещение к интернализации ответственности.

## **Не бросай себя в глубину**

Одна из самых распространённых и предотвратимых ловушек — браться за самые продвинутые задачи и отказываться отступить к лучше откалиброванному уровню, когда возникают трудности.

Для подавляющего большинства людей прыжок головой вниз без baseline foundational ability — не жизнеспособный подход.

---

## **ПУТЬ**

### **Не зацикливайся на юношеских соревнованиях**

Ну и что, что ты не выиграл [соревнование для молодёжи здесь]? Ещё не поздно оставить след. Юношеские соревнования — просто учебные арены для early bloomers. Они не являются gatekeepers твоего будущего.

В долгосрочной перспективе долгосрочная игра — единственная, которая имеет значение, и пропуск краткосрочной игры не означает, что ты упустил свой шанс.

## **Слишком легко переоценить своё значение через percentile**

«Вау, я в 99-м percentile, я такой крутой...» — нет, ты один из 100. В стране со 100 миллионами людей таких как ты — миллион. Если хочешь быть одним на миллион — нужно улучшиться на несколько порядков величин.

Если хочешь иметь точное интуитивное ощущение того, насколько ты хорош — думай не о своём percentile. Думай о том, сколько таких как ты.

## **3 стадии развития таланта**

Это обнаружил Benjamin Bloom в 1980-х, изучая историю тренировок 120 world-class талантливых людей в 6 talent domains (фортепиано, скульптура, плавание, теннис, математика, неврология):

**Стадия I: Веселье и увлекательное playtime.** Студенты только начинают развивать awareness и интерес. Учитель даёт обильную положительную обратную связь и поощряет исследование. Студентов вознаграждают за effort, а не за achievement.

**Стадия II: Интенсивное и напряжённое skill development.** Студенты полностью сосредоточены на улучшении performance. Учитель превращается в коуча, сосредоточенного на тренировочных упражнениях. Критика конструктивна; положительная обратная связь даётся за achievement.

**Стадия III: Развитие собственного стиля и раздвигание границ области.** Студенты центрируют всю жизнь вокруг talent domain и работают с world-class экспертом.

Failure modes:

- **Вечный студент:** perpetually избегает прыжка в creative production.
- **Wannabe:** прыгает к creative production до того, как foundational skills готовы.
- **Дилетант:** никогда не выходит за пределы playtime, никогда не commit'ится к серьёзному foundational skill development ни в чём.

Все хотят добраться до стадии 3, но никто не хочет grinding через стадию 2. Нет стадии 3 без стадии 2.

## **В развитии таланта нет shortcuts**

Если Эйлер, Гаусс, Максвелл, даже Рамануджан, Галуа и т.д. должны были спин-апиться на foundational knowledge — так же и ты.

Когда кажется, что кто-то прогрессирует так быстро, что «должен» был взять shortcut — на самом деле он speed-ran foundations.

## **Потеря интереса иногда связана с отсутствием foundations**

Если теряешь интерес к своим целям — возможно, сомневаешься, что можешь их достичь. Если сомневаешься — возможно, flailing'ешь. Если flailing'ешь — возможно, пытаешься взяться за слишком большую starting challenge относительно своих underlying skills.

## **Если делаешь глупые ошибки — нужно больше практики**

Подъём по skill hierarchy — это не только conceptual understanding, но и reliable execution. Если делаешь «глупые ошибки» — нужно больше практики, и всё.

## **No train, no gain**

Если не измеряешь performance и не предпринимаешь действий для его улучшения — не тренируешься серьёзно, просто играешь.

### **Почему стоит push'ить себя**

Потому что дорога всегда тянется дальше, чем видно перед тобой, и ты максимизируешь вознаграждение, путешествуя как можно дальше. Push'ить себя — это избегать стагнации. Эффективно выращивать skills и постоянно конвертировать их в новые growth experiences.

### **Держи ногу на газу**

Когда что-то поначалу даётся легко — соблазнительно убрать ногу с газа и попытаться откатиться оставшийся путь. Откатываешься какое-то время, пока это не становится новой нормой, пока не забываешь, что есть газовая педаль. И в итоге останавливаешься прямо над основанием горы.

### **Ты — автомобиль**

Ты — автомобиль. Быстро едешь по асфальтированным дорогам и застреваешь в грязи. Прокачай двигатель до максимально быстрого/мощного, старайся оставаться на дорогах и избегать грязных зон. (Тренируй strengths в superpowers и адаптируй окружение и цели под них.)

Но одновременно — не позволяй немного грязи (weakness) дерейлить твоё путешествие.

### **Что делать, когда упираешься в ceiling**

В многих talent domains upskilling в какой-то момент становится трудным и неинтуитивным для всех. Это создаёт не жёсткий потолок, а мягкий — точку, в которой количество времени и усилий для дальнейшего прогресса начинает стремительно расти.

Когда упираешься в ceiling — решение, как правило, состоит в pivot'e в направлении, где потолок выше.

### **Легче любить игры, для которых ты создан**

Оценивай, где твой ceiling наиболее высок, и учись любить игры, для которых ты создан.

Как однажды перефразировал читатель Abhishek Singh: «Meta-skill — знать, какая игра заслуживает твоих тренировок на данном этапе жизни, и накапливать прошлый опыт так, чтобы начинать не с нуля, а на пересечении, где мало кто может конкурировать.»

### **Compound усердный труд и удачу**

Как усердный труд, так и удача необходимы для величия, и ни одно из них не является достаточным само по себе. Однако они переплетены и питают друг друга:

- Если работаешь усердно — с большей вероятностью получишь удачу: создаёшь больше «luck surface area» и лучше подготовлен к её использованию.
- Если используешь удачную возможность и чувствуешь, что усердная работа даёт результат, — воодушевляешься и мотивируешься работать ещё усерднее.

Величие возникает из virtuous cycle усердного труда и удачи, compound'ящих друг друга.

### **Поставь себя в позицию, где можешь рисковать**

Недооценённый компонент career fit — накопить достаточно сбережений для преследования возможностей с неопределённым вознаграждением. Все знают, что

обучение — огромная компонента карьерного успеха, но так же важна способность «eat risk». И это также огромное конкурентное преимущество.

### **Привяжи свой комфорт к реальной, осязаемой ценности**

Весь смысл skills, технологий, суперсил — совершать более великие вещи. Не придерживаться статус-кво с меньшей работой.

Способ избежать ловушки — развить сильные эмоциональные связи с высокой проблемой, где её постепенное решение постепенно трансформирует жизнь людей, которые тебе важны, одновременно делая твою жизнь комфортнее.

### **Если не можешь найти захватывающую работу...**

Если не можешь найти работу, которая по-настоящему захватывает, — начни (публично) делать своё дело. Ты встретишь других людей, делающих интересные вещи, и всегда можешь переоценить, хочешь ли продолжать самостоятельно или объединиться с ними.

### **Как распределять bandwidth в поиске миссии**

Если ещё не нашёл единственную миссию — рекомендуемое распределение:

- Один **главный фокус** (нагрузка эквивалентна full-time работе)
- Один **полу-фокус** (нагрузка эквивалентна part-time работе)
- Всё остальное — **хобби** с оставшимся временем

---

## **КОМАНДА**

### **Если просишь кого-то быть ментором — делаешь это неправильно**

Способ №1 увеличить продуктивный output — иметь его вытянутым из тебя более старым опытным человеком, который unreasonably demanding, невероятно supportive и имеет твоё глубочайшее уважение.

Но вот в чём загвоздка: чтобы найти такого человека, быть достойным его времени и иметь этот дополнительный output вытянутым из тебя — как правило, нужно прежде всего быть невероятно усердным и талантливым работником.

Если просишь кого-то быть твоим ментором — делаешь это неправильно. Это должно выглядеть меньше как они помогают тебе и больше как ты помогаешь им.

### **Создай давление на своего начальника**

Один из лучших карьерных hacks — выполнять работу так быстро и хорошо, что создаёшь давление на начальника придумывать тебе больше работы. Это заставляет проекты расти в масштабе, сложности и ответственности.

### **Попади в правильную команду**

В правильной команде твоя усердная работа вдохновляет товарищей по команде повышать свою игру. В неправильной — твоя усердная работа заставляет их откидываться назад и делать меньше.

### **Конкуренция как форма сотрудничества**

Конкуренция и сотрудничество звучат как противоположности, но есть способ, которым конкуренция может разумно рассматриваться как форма сотрудничества. Это когда engaged в дружеской конкуренции с людьми, с которыми у тебя связь и которые тебе

важны, где смысл — мотивировать друг друга и делать друг друга лучше.

### **Твоя цель — НЕ доказать, что ты умный**

Когда работаешь в команде — основная цель не доказать, что ты умный. Это заставить проблемы исчезать.

Не получаешь очков за создание ненужной проблемы для демонстрации своего интеллекта. Не получаешь очков за overcomplicated решение простой проблемы. За это теряешь очки. Очки получаешь за то, что берёшь проблему и заставляешь её полностью испариться.

### **Произведи первое впечатление вкладом, а не критикой**

Когда присоединяешься к новой команде — произведи первое впечатление вкладом, а не критикой. Мудрее разобраться в окружающем контексте, продемонстрировать способность и готовность внести вклад, и ПОТОМ предложить улучшение.

### **Никогда не приходи с пустыми руками**

Когда tasked с решением чьей-то проблемы и дела идут не очень хорошо — лучше вести log всего, что пробовал или исследовал, как лабораторный дневник.

Даже если не можешь решить проблему, перечисление множества «вот что я пробовал и вот что произошло» может побудить руководителя предложить следующий шаг. И это демонстрирует честный труд.

### **Тебе нужен берсерк во главе**

Прогресс всегда зависит от небольшого числа людей, работающих heads down над десятилетней миссией и серьёзно держащих линию. Это постоянная борьба с посредственностью, и без берсерка во главе и герметичной chain of accountability — ничего не достигнешь.

---

## **МИССИЯ**

### **Выбрать хорошую задачу для работы**

Над какой задачей работать? Поначалу — неважно. Просто работай над любой toy problem, которая тебя интересует, чтобы строить технические skills и получать domain knowledge.

Но как только начинаешь углубляться в задачи, требующие многих лет full-time работы, выбор хорошей задачи становится очень важным.

Два failure mode'а плохих задач:

**Failure Mode 1:** Нет реализуемого видения решения. Задача слишком сложна.

**Failure Mode 2:** Людям не важна задача. Они не готовы платить за решение. Значит, не будет вознаграждения за решение задачи.

Как найти задачу, избегающую обоих failure mode'ов? Нужно найти (или создать) пересечение своих интересов/талантов, области возможного и желаний внешнего мира.

### **Чтобы понять, что тебя fulfills — нужно работать очень чертовски упорно**

Решение «не знаю, что хочу делать» — НЕ «тогда, наверное, ничего не буду делать». Даже при полном отсутствии self-knowledge — просто попробуй кучу случайных активностей и поразмысли, что понравилось, что нет.

Self-knowledge — не часть базовой установки. Ты не spawn'ишься с ним. Нужно упорно работать, чтобы приобретать его по кусочку, упражнение за упражнением, опыт за опытом.

### **Формула «Progress Equals Pressure»**

Большинство знают, что нужны hardcore skills для hardcore вещей. Но многие даже с hardcore skills не знают, что также нужно защищать себя от nerd-sniping ([xkcd.com/356](http://xkcd.com/356)).

Прогресс — о давлении на задачу: применении силы своих skills к конкретной области (давление = сила / площадь).

### **Люби то, что делаешь**

Если хочешь дико успешного результата и считать это хорошо прожитой жизнью — нужно любить то, что делаешь. Другого пути нет.

Любовь — это постоянные hardcore усилия. Не как описание, а как определение. Последовательно hardcore люди достигают экстраординарных результатов через экстраординарные действия.

### **Будь builder, а не только fighter**

Когда приобретаешь hardcore skills и открываются новые карьерные возможности — старайся избегать тех, что ставят тебя против других hardcore-skilled людей в zero-sum играх. Хочешь создавать новую ценность, а не просто бороться за существующую.

### **Строй там, где строительство создаёт больше возможностей**

Никто, кто строит что-то с высокой будущей ценностью, не беспокоится, что AI займёт их работу. В такой обстановке building расширяет твою opportunity surface area строить ещё более ценные вещи. Чем больше делаешь — тем больше остаётся делать.

---

## **МОТИВАЦИЯ**

### **Состояние «Blah»**

Знаешь, как когда проводишь весь день на диване, смотришь TV, становишься каким-то беспокойным, но как-то и слишком усталым, чтобы встать? Это состояние «blah».

То же самое происходит с обучением. Если смотришь кучу лекций, документалок, учебников, не решая активно задач — впадёшь в то же состояние «blah».

Единственный выход из «blah» — начать активно делать вещи, требующие труда. Удовлетворение приходит от преодоления и достижения того, что требовало реальной работы.

### **Безразличие — часто просто overwhelm**

Поразительно, сколько людей в итоге наслаждаются [трудным делом], когда задачи правильно откалиброваны к их текущему уровню. Никто на самом деле не сходит с upskilling-поезда из-за потери интереса. Сходят, потому что стагнирующий прогресс заставляет сомневаться, что поезд когда-либо доберётся до цели.

## **Часто можно просто заставить себя полюбить вещи**

Многие думают, что их likes и dislikes фиксированы. Но правда в том, что часто можно просто заставить себя полюбить вещи.

Поразительно, каким fun может стать кажущееся скучным занятие, когда выработаешь привычку, установишь baseline компетентность и получишь skin in the game.

## **Почему extrinsic motivation важна**

Принято думать, что «учиться ради самого обучения» лучше, чем «учиться ради достижения extrinsic цели» — но я не думаю, что это правда.

Оптимальная мотивация требует баланса intrinsic и extrinsic факторов. Intrinsic мотивация заставляет работать над интересными вещами с уникальной перспективой. Extrinsic мотивация держит тебя на рельсах долгосрочных целей.

Intrinsic vs extrinsic — ложная дихотомия. Нет предела тому, сколько мотивационных факторов ты можешь накапливать в каждой категории — и нет предела тому, насколько мотивированным можешь быть.

## **Как стать super-producer**

Trick №1 к super-продуктивности — interleaving широкого разнообразия продуктивной работы, которая тебе нравится. Устаёшь, скучаешь и становишься непродуктивным, если слишком долго движешься в одном направлении.

Решение — оказаться в ситуации, где твоя production function имеет многомерный input, и все эти inputs имеют large partial derivatives примерно одного порядка величин. Следуешь gradienty в этом пространстве, cycling между этими activities.

## **Как top performers поддерживают огромную нагрузку**

В большинстве областей самые талантливые performers имеют edge, позволяющий level up'иться быстрее среднего. И несмотря на этот edge — они всё равно выбирают вкладывать намного больше времени, чем средний человек.

Способ поддерживать такой высокий объём работы — interleaving широкого разнообразия продуктивных активностей.

## **Преодоление парадокса серьёзных тренировок**

Одна из hallmark серьёзных тренировок — твоё прошлое я считало бы нынешнее я высокой или даже недостижимой целью. Но в краткосрочной перспективе ощущение совершенно иное. День за днём можешь не видеть этого, можешь сомневаться, что это происходит.

Способ, которым становишься сильнее — постоянно поднимать достаточно тяжёлые веса, чтобы чувствовать себя слабым. Никогда не перестаёшь быть сломленным тренировкой, и основное ощущение, которое испытываешь день за днём — слабость, а не сила.

Вот почему так важно периодически сравнивать возможности нынешнего себя с прошлым.

## **Радость от upskilling**

Радость от upskilling похожа на радость от физических упражнений. Нужно её заработать. Упражнения становятся намного веселее, когда замечаешь начало

трансформации тела. С upskilling — то же самое.

Прогресс вдохновляет мотивацию и радость — не наоборот. Не недооценивай, сколько мотивации и радости ощутишь, когда сможешь делать упражнения, которые раньше не мог.

### **Ложь о том, что обучение должно быть приятным**

Многим студентам продают ложь, что обучение должно ощущаться приятным. И когда это не их опыт — cognitive dissonance их пожирает.

Что скажет хороший тренер: «Да, это ощущается как тяжёлая работа, и это нормально. Наша цель здесь — не удовольствие, а удовлетворение: делать прогресс, делать вещи, которые раньше не мог, хорошо себя чувствовать от достигнутого. Ты здесь, чтобы работать — значит, работаем.»

### **Как изнурительная работа становится fun**

Поначалу у тебя куча недостающих foundational knowledge. Жёсткое пробуждение. Потом просто фокусируешься на одном кусочке за раз. В конце концов становится немного fun. И в какой-то момент оглядываешься и понимаешь, что трансформировался.

Это как прийти в спортзал слабым, не особо желая тренироваться, но просто сделать это и придерживаться плана — и в итоге привыкнуть и реально кайфовать, кладя серьёзный вес на гриф. Fun не в смысле «это так приятно», а в смысле «это заставляет меня чувствовать себя legit».

### **Зачем стараться, если другие могут сделать лучше?**

Если ты не самый умный, самый high-agency человек, которого знаешь — смысл делать что-то, что они могли бы сделать быстрее и/или лучше?

Скажу тебе. Даже самые умные, самые high-agency люди в мире severely bandwidth-constrained и не добираются до большинства вещей, которые потенциально могут сделать. Если ты делаешь что-то, что кто-то другой мог сделать быстрее, но не сделал — реальность такова: ты это сделал, а они нет.

---

## **ОБУЧЕНИЕ**

### **Величайший educational life hack: учиться заранее**

Зачем учиться заранее? Потому что это защищает от многих академических рисков, открывает все виды карьерных возможностей и позволяет войти в эти двери раньше в жизни.

Когда pre-learn'ишь материал курса перед его прохождением, практически гарантирована пятёрка. Ты защищён от всех возможных рисков: слишком быстрый темп курса, плохие объяснения, пропущенные prerequisites и т.д.

Зачем останавливаться на одном годе вперёд? Стоит продолжать и дальше ускоряться. Дорога всегда тянется дальше, чем видно перед тобой.

### **Зачем быть намного впереди? Это покупает время.**

Для всех, кто заинтересован в STEM-карьере, раннее приобретение продвинутых технических skills открывает самую ценную вещь в существовании. То, что даже деньги

не могут купить. ВРЕМЯ.

Есть гонка, но то, против чего ты гонишься, намного страшнее и мощнее, чем любой другой человек. Ты гонишься против ВРЕМЕНИ САМОГО.

Pink Floyd выразили это лучше всего: «Ты молод и жизнь длинна, и сегодня есть время убить. И потом однажды обнаруживаешь, что десять лет остались позади. Никто не сказал тебе, когда бежать. Ты пропустил стартовый выстрел.»

### **В чём смысл instruction**

Серьёзный upskilling открывает дверь к вещам, о которых мечтал. Задача teacher'a — найти самый пологий путь на гору и построить лестницу по этому пути. Тогда студенту нужно просто сосредоточиться на каждом последовательном шаге.

### **Когда происходит обучение?**

Обучение — постепенный gain в способности выполнять tangible, воспроизводимый skill. Если не получаешь таких gains — не учишься.

Представь теннисные уроки с личным коучем. Когда происходит обучение? Не когда платишь коучу деньги. Не когда смотришь, как коуч демонстрирует движение. Когда начинаешь делать то, что раньше не мог.

### **Не существует low-effort обучения**

Talent development требует работы — не просто немного, а много. Эффективное обучение должно центрироваться вокруг deliberate practice.

К сожалению, восприятие обучения часто противоречит реальному измеримому обучению. При использовании эффективных стратегий студенты показывают лучшие результаты, но могут чувствовать, что узнали меньше.

Многие методы training неэффективны, но требуют мало усилий и создают иллюзию понимания — например, смотреть на заметки, перечитывать материал, подчёркивать.

### **Величайший прорыв в науке об обучении**

Величайший прорыв в науке об обучении за последний век — характеристика механики обучения в мозге.

Обучение — это всё о взаимодействии между working memory (WM) и long-term memory (LTM).

Цель обучения — увеличить количество, глубину, retrievability и обобщаемость концептов и skills в LTM.

WM имеет строго ограниченную capacity — большинство может удерживать только около 4 «chunks» одновременно, лишь около 20 секунд. Эта ограниченная capacity делает WM bottleneck'ом при transfer информации в LTM.

Когда достаточно освоил skill — влияние WM на выполнение задачи снижается, потому что information processing перенесён в LTM.

Но даже при всём правильном — нужно также справляться с забыванием. Решение — review через активное retrieval из LTM.

### **«Following along» против настоящего обучения**

Принято думать, что «следовать» = учиться. Но это не так. Это ощущение полностью искусственное.

Обучение — позитивное изменение в долгосрочной памяти. Не учился, если не можешь последовательно воспроизводить потреблённую информацию и использовать её для решения задач.

Комфортная fluency при следовании возникает из-за того, что окружающий контекст уже у тебя в голове — тебя не заставляют извлекать его из LTM.

### **Необходимость практиковать retrieval из памяти**

Если хочешь реально удержать потреблённую информацию — нужно практиковать retrieving её из памяти, а не просто повторно потреблять.

Ты удержал информацию тогда и только тогда, когда можешь воспроизвести её из памяти.

Retrieval — это специфическое действие «вытаскивания» информации из LTM в WM. Если загружаешь информацию в WM, глядя на reference material, вместо вытаскивания из LTM — не укрепляешь retention. Как идти в тренажёрный зал поднимать веса, но давать спотер'у делать всю работу.

### **Порочный круг забывания**

Когда постоянно смотришь назад на reference — информация не остаётся в мозге. Reference material становится костылём.

Ты в порочном круге: смотришь на reference, потому что не помнишь. Не помнишь, потому что не переносишь в LTM. Не переносишь, потому что не практикуешь retrieval из памяти. Не практикуешь retrieval, потому что всегда смотришь на чёртовый reference!

Единственный выход — retrieval practice. Всегда старайся изо всех сил вспомнить информацию из памяти, ПРЕЖДЕ чем посмотреть её. Если не получается после всех усилий — нормально ненадолго заглянуть — но только как последнее средство. Никогда не копируй из reference.

### **Один из худших ошибок при учёбе**

Один из худших ошибок — рефлекторно посмотреть назад на reference, не попытавшись сначала вспомнить из памяти. Это как идти в тренажёрный зал и давать спотер'у поднять вес в момент, когда становится тяжело.

### **Сначала retrieval, потом рассуждение**

Если хочешь строить automaticity — нужно практиковать автоматический recall. Не давай себе рассуждать/выводить результат перед его применением. Применяй сначала — выводи/обосновывай потом.

### **Порочный круг context overload**

Один из наименее эффективных способов учиться — сразу бросаться на самый сложный «аутентичный» контекст. Намного эффективнее разобрать skills до простейшего возможного контекста, сделать несколько repetitions и постепенно наращивать сложность.

### **Prereq Yo' Self Before You Wreck Yo' Self**

Быть за пределами своей глубины в skill training — огромная проблема, потому что learning progress останавливается. Это не поезд, который отправился поздно. Это поезд, который вообще не движется.

Единственный способ снова двинуться вперёд — опуститься и работать на уровне, подходящем тебе.

### **Заполнение foundations — разница между ямами и пропастями**

Если заполнишь свои foundations, пропущенные prerequisite знания — сможешь заполнить шаги, пропущенные в крутых технических книгах и проектах. Заполнение foundations — разница между этими пропущенными шагами как незначительными ямами против непреодолимых пропастей.

### **Планируй путь top-down, выполняй шаги bottom-up**

Top-down подход может быть полезен для планирования broad-strokes learning journey к цели. Например, думая top-down, можно выяснить, какие математические поля нужно учить, чтобы ML стал доступным.

Но granular шаги путешествия, само обучение, нужно выполнять bottom-up.

### **Эффективный learning loop**

Если хочешь максимизировать learning efficiency — хочешь streamlined instruction, большинство времени на активное problem-solving и быстрое переключение между instruction и problem-solving.

Это постоянный цикл минимальных эффективных доз: минимальная эффективная доза чёткого instruction → минимальная эффективная доза problem-solving → назад к instruction для немного более сложных проблем → и так далее.

### **Не раздувай feedback loop**

Многие слышат «повторяющийся» как синоним «бездумного», тогда как repetition и mindlessness полностью ортогональны. Хочешь «mindful repetition» — mindful, потому что практикуешь прямо за пределами своего repertoire, получая feedback при каждой попытке.

### **Ловушки при обучении через проекты**

Когда приобрёл foundational skills для крутого проекта — вперёд! Но вот ловушки:

- Не используй проекты для приобретения fundamental skills — обычно неэффективно.
- Убедись, что проекты guided. Не ставь себя в позицию, где крутишь колёса впустую.
- Не позволяй проектам слишком сильно вгрызаться в foundational skill-building.

### **Повторение должно ощущаться как вызов**

Студенты часто ожидают, что review будет лёгким. Но при максимизации learning efficiency — reviews должны ощущаться сложными.

Именно акт успешного извлечения нечёткой памяти (а не чёткой) продлевает срок памяти. Spaced repetition = «wait»-lifting.

### **Учись как поднимаешь штангу**

Spaced repetition настолько похож на weight training, что можно назвать его wait training. Поднимаешь информацию с пола LTM и поднимаешь её в WM. Чем нечётче информация,

тем тяжелее поднять. Ожидание создаёт вес.

### **Schooling против talent development**

Принципы эффективного training похожи между доменами. Но видишь это только если реально оптимизируешь performance.

В schooling студентов группируют прежде всего по возрасту, а не по ability, и каждая группа в lockstep проходит curriculum. В talent development студенты проходят через skills с разной скоростью, но учат skills до одного порога performance.

### **Sanity check для эффективных методов обучения**

Каждый раз, когда учишься, представь, что Жнец явится в конце сессии, чтобы проверить тебя по пройденному — и если не ответишь правильно на любой вопрос, умрёшь. Какие учебные техники ты бы использовал в этой ситуации, уже должен использовать.

---

## **ЭКСПЕРТИЗА**

### **Движущая сила экспертизы — долгосрочная память**

Важный механизм экспертной performance — «perceptual learning»: способность извлекать ключевые features из сложных окружений, фильтруя нерелевантный шум.

Тогда как beginner воспринимает отдельные изолированные куски информации — expert воспринимает «chunks» информации, организованной в осмысленные паттерны и структуры. Эти chunks физически закодированы как проводка в LTM эксперта.

Более того, эти memory representations в мозге эксперта могут также включать predictive информацию, которой нет в оригинальном stimulus.

### **Обучение — это память**

Память — сырой материал понимания, и мы используем её для построения великолепных информационных структур в мозге.

Глубокое понимание состоит не только из декларативных фактов, но и из connections, которые связывают факты в related группы или «chunks», connections, которые связывают меньшие chunks в большие, а также procedures для работы с chunks.

«Обучение — это память» может казаться очевидным, но многие learners не в полной мере понимают импликации. Если не понимаешь, что обучение — это память, не будешь понимать, что самый эффективный способ учиться — использовать memory-supporting training techniques.

### **Обучение — это строительство мостов, а не прыжки**

Есть lack of research evidence, что можно увеличить «сырую» WM capacity и generalization ability через training general problem-solving skills.

Но ЕСТЬ research evidence, что можно эффективно превратить LTM в extension of WM, если приобрести domain-specific foundational skills и развить их до automaticity.

Способ улучшить способность делать mental leaps — не прыгать дальше, а строить мосты, которые сокращают расстояние для прыжка.

## **Всё сводится к domain knowledge**

Нет доказательств, что можно увеличить способность кого-то к problem-solving любым другим методом, кроме оснащения его бóльшим domain knowledge.

## **Твой мозг должен тебя удивлять**

Если тебя периодически не удивляют возможности собственного мозга — ты, вероятно, тренируешь его неправильно. Твой мозг — самая мощная machine learning model, которую ты когда-либо будешь использовать.

## **Превращай магическое в механическое**

Когда процесс или явление ощущается магическим — это обычно указание, что ты на самом деле не понимаешь, что происходит под капотом.

Сдвиг от магического к механическому не умаляет красоты. Он просто меняет линзу. Перестаёшь быть зрителем чудес и начинаешь становиться их строителем.

---

*Оригинальный текст Джастина Скайкака, опубликован 12 января 2025 года:  
[justinmath.com/advice-on-upskilling/](https://justinmath.com/advice-on-upskilling/)*

*Русский перевод создан 2 июня 2026 года. Английские термины — upskilling, skill tree, deliberate practice, the grind, feedback loop, working memory (WM), long-term memory (LTM), spaced repetition, skill stacking, ROI, STEM, alpha, virtuous cycle, domain expertise — сохранены в оригинальном виде.*