

АЛЬПИНА.ПЛЮС × ЖЕМАЛ ХАМИДУН

ТРАНСКРИПТ

ИИ для (не)разработчиков

2025-06-20 · 1 ч 40 мин · С Сергеем Андрияновым · Бесплатный вебинар

Расшифровка аудио: Deepgram Nova-2, русский язык, уверенность распознавания 96.6%.

Абзацев: 136 · Длительность: 1 ч 40 мин

00:00:00 Давайте тогда стартовать. Мы сегодня такую тему очень интересную решили поднять, которая сейчас я на следующий слайд перейду и раскрою ее. А пока что еще раз вас призову. Если вы еще вдруг не с нами, не на нашем канале дело в promptе, то welcome in board. У нас там много интересного регулярно обсуждается, там есть возможность в комментариях писать, и вообще очень интересные кейсы применения нейросетей рассматриваются.

00:00:29 Так что, в общем, welcome, подписывайтесь или отсканировав QR-код, или перейдя по ссылке, которую я еще разочек продублирую. Сегодня мы рассмотрим тему искусственный интеллект для не разработчиков. Тут специально не отдельно в скобках, потому что мы сегодня посмотрим на тему применения искусственного интеллекта в написании кода для каких-то сервисов, приложений, в общем, много для чего. И посмотрим на нее через 2 грани, через 2 призмы. С 1 стороны, через призму разработчика с большим опытом и технического архитектора, а с другой стороны, через призму не инженера, не разработчика, через мою, когда я пробовал тоже кодить некоторые вещи с помощью искусственного интеллекта и получалось, в общем, интересно и неплохо.

00:01:23 Поэтому с 2 сторон на это все посмотрим и со стороны разработки, и со стороны не разработки. Почему мы сегодня, собственно, можем об этом рассказать? Если вдруг мы не знакомы еще, то представлюсь тоже. Меня зовут Джемал, я директор по цифровым продуктам Alpina Digital. Собственно, был 1 из зачинателей AI-трансформации в Альпине.

00:01:46 Мне кажется, это такой процесс неостановимый и очень длинный. С 23 года мы уже занимаемся тем, что внедряем в разных частях альпины искусственный интеллект. Вот Alpina GPT родилась как платформа Благодаря этому, потому что мы внутри искали способы применения нейросетей, передачи сотрудникам инструмента, который удобный доступ ко всем нейросетям давал бы в 1 месте. Сейчас мы вплотную подошли к тому, чтобы внедрять искусственный интеллект в разработку, в кодинг у нас в Alpine. По мотивам сначала внутреннего такого вебинара родилась вот эта тема, которую мы решили вынести во вне и вообще с вами тоже поделиться этим, поговорить об этом и, возможно, какой-то ваш фидбэк получить на это.

00:02:35 И с нами сегодня, кстати, у меня есть канал, где я тоже пишу регулярно про нейросети, тоже подписывайтесь на него, ставьте лайки и все такое. С нами сегодня Сергей Андрьянов, мой друг, партнер и сетевой фаундер студии

разработки Regul, наш партнер по разработке сложных IT-решений. Если вы сегодня из приложения Alpina Plus, то вы из приложения, которое разрабатывает команда Сергея. Если вы пользуетесь новой версией Alpina GPT, то вы тоже пользуетесь приложением, которое разрабатывает команда Сергея. Вот.

00:03:11 Серег, может, если что-то добавишь еще, я могу только скромно сказать, что ты более 20 лет имеешь опыт разработки инженерных практик. Может быть, что-то еще хочется докинуть к этому. Я думаю ты прекрасно все сказал, но мне добавить нечего. Я сохраню скромность. Да, Сергей скромный парень.

00:03:34 Хорошо, у нас сегодня с вами продолжительность лекций около часа и примерно 30 минут на ответы, вопросы, которые будут после выступления. У нас такой с вами живой формат сегодня. Помимо того, что будет презентация, какая-то лекционная часть, мы с вами еще можем повзаимодействовать, поотвечать на ваши вопросы, возможно, что-то показать, потому что тема, которая сегодня рассматривается, она такая во многом очень практическая. Вопросы вы пишете здесь атавизм из МТС-линка, вопросы пишете в разделе платформы Вопросы. Этого здесь нет, поэтому просто пишете их в чат.

00:04:10 Мы их будем ловить и стараться на них ответить. Кто будет с нами до конца сегодня получит подарок доступ к платформе Alpina GPT на 7 дней абсолютно полный, бесплатный. Вы сможете все нейросети использовать в 1 месте, даже попробовать покодить что-то по мотивам сегодняшнего мероприятия. Давайте с вами сверимся немножко и попробуем как-то выровняться, да, немножко используете ли искусственный интеллект, как часто используете. Напишите, пожалуйста, очень интересно каждый день.

00:04:47 Большими буквами точно каждый день. Спасибо большое! Каждый день. Еще у нас сегодня продвинутая аудитория будет. И да, как раз возможность потом поприменять это, то, о чем мы сегодня расскажем, поприменять это в практике.

00:05:07 Так, каждый день, каждый день, теперь каждый день. Да, да, Александр, спасибо. Да, некоторые участники сегодняшнего вебинара это участники, в том числе, и с нашего курса, который проходил совсем недавно, практически завершился поток. Сегодня была последняя такая годовая конструкция. Вообще, у нас перезапускается поток, будет в июле, но отдельно чуть попозже про это расскажу в конце.

00:05:35 Так, хорошо. Выровнились чуть-чуть. В общем, у нас аудитория кажется продвинута. Давайте небольшой такой блиц-опрос. Вот тут на экране 4

картиночки.

00:05:45

Это картинки из игр. И вопрос звучит следующим образом: на ваш взгляд, какая из игр сделана человеком, а какая сделана искусственным интеллектом? Я специально взял такие простенькие игры, которые вот когда-то, не знаю, на заре использования компьютеров или вот типа Minecraft, да, то, что сейчас очень популярно среди подрастающего поколения, вот. То есть, сейчас уже можно делать игры с помощью ИИ. Вот интересно, какие на ваш взгляд картинки игр, которые сгенерированы искусственным интеллектом.

00:06:24

Вот Георгий написал третьи И, 1 и 3 и 1 и, 1 и, 1 и, 1 и 3 и, так вторая сделала И, 1 и 3 и в целом, традиционно, когда мы какие-нибудь картинки показываем, мнение всегда разделяются, да, 2 и 3. Супер, не буду вас томить. На самом деле, очень интересно, но, в целом, вот простые игры уже делаются с помощью ИИ, зачастую, даже совершенно не хуже, чем делаются людьми. Картинки 1 и 2 это игры, сгенерированные искусственным интеллектом через такую платформу оазис, значит, 3-4 это вот игры, созданные человеком. Но, на самом деле, пока что вот такие простые, специально взял игрушки несложного уровня, они уже довольно легко кодаются с помощью и буквально за несколько минут, фактически, можно что-то простенькое развернуть, какой-то тестовый вариант.

00:07:31

Конечно, более сложные игры это следующий этап, но вот даже после сегодняшнего вебинара, я думаю, что вы можете пойти и попробовать тестово что-то для себя такое создать простенькое, что возможно вас сподвигнет к тому, чтобы еще глубже в тему и кодинг погружаться. В этом месте, примерно, Серег, я, наверное, свою демонстрацию остановлю и передам тебе слово. Хочется, чтобы ты, наверное, поделился, как раз, как же стать разработчиком, не будучи разработчиком. Войти, войти с помощью ИИ. Да, что-то типа того.

00:08:12

Так, сейчас я сделаю демонстрацию. Да, видим презентацию. Пожалуйста, напишите в чатик, видно ли вам презентацию. Кстати говоря, презентация тоже была сгенерирована с точки зрения написания кода. Вот это прям страничка отдельная, которую сделали с помощью, ну в общем, Серег, тебе слово.

00:08:37

Я хотел этим подытожить, но раз ты проспойдивил, да, уже скажем. Понял, челлендж. Вся эта презентация, да, она была сгенерирована буквально минут за 20 с помощью и Cloud Appus 4. Да, контент, соответственно, подготовили заранее, но это было 20 слайдов, довольно-таки перегруженных информацией. Нейронка обработала и информацию, соответственно, и

перефразировала, и сделала текст более лаконичным, и придумала дизайн, и сверстала, в том числе даже с анимацией, я не буду сейчас показывать, но даже адаптивно для телефонов.

00:09:17

Единственный момент, который мне пришлось более трех раз ей уговаривать ее поправить, это вот в этом месте ехал текст, и нейронка пыталась забивать гвозди микроскопом, с трех попыток у нее не получилось, мне пришлось ей конкретно сказать, что поставь такое-то CSS-свойство, это единственная сложность, которая у нее возникла, остальное буквально в пару итераций, в пару правок за 20 минут все было сгенерировано. Поэтому это вот можно сказать первый кейс, то есть вот подобные вещи, лендинги, все уже можно сейчас сделать без программистов, без дизайнеров, в принципе если нужно что-то быстрое, быстро, дешево, сердито, и в принципе мне кажется даже симпатично. То есть тот кейс, где в принципе и может разминить разработчика, но в целом я бы не сказал разработчика, скорее вебмастера. Ну и можно продвигаться дальше, соответственно о чем мы сегодня поговорим. Ну, злободневный вопрос, который задают все и все время может ли заменить или не может, в принципе, нажимал ты мою позицию, знаешь, ты ко мне с этим вопросом приходил еще год назад, я тебе тогда сразу точно говорю, нет, не может и никак, потом осенью прошлой мы с тобой разговаривали, и у меня уже был не такой уверенный голос, сейчас мы об этом поговорим, но да, как бы уверенности в голосе меньше становится постепенно, но об этом дальше.

00:11:03

С чего начать использование AI, ну тут я думаю мы с тобой поговорим совместно, ты как не программист, я как все-таки больше к технической стороне относящийся человек, и также дам какие-то практические рекомендации, где можно улучшить, так называемый, сейчас часто звучащий, вайп-кодинг. Ну и какие подводные камни, риски ждут, чего стоит остерегаться, и где стоит ну очень внимательно смотреть и не доверять, чтобы не попасть в неприятные истории. Здесь я даже подготовил интересный кейс. Так, ну мы также поговорим о некоторых инструментах и концепциях современных, то есть таких как depresearch, очень крутая вещь, без которой я уже даже не представляю как жить. Cloud Code, как 1 из примеров, как раз таки, как не программисту можно что-то разрабатывать, МЦП, Model Context Protocol, очень нашумевшая сейчас аббревиатура, многие возможно про нее слышали, далее о ней расскажу подробно, но на мой взгляд действительно, как бы, это можно сказать прорывная технология, как в принципе как сами ее изобретатели компании Антропик говорят, USB для искусственного интеллекта.

00:12:47

Покажу несколько практических кейсов с примерами, и в конце поговорим как раз таки об этих злободневных этических вопросах и безопасности, где

стоит, а где не стоит. Ну, в целом какие преимущества нам может дать ИИ, ну помимо того что классика жанра, которую уже сейчас в принципе наверное используют практически все, возможно кроме бабушек и дедушек, это чаты, вопросы, вместо Google, так сказать, расширение своей экспертизы без ее реального расширения. Да, это в принципе есть, тут можно сильно не останавливаться, все про это знают, единственное, что скажу, нельзя доверять и работать с этим как копировать, вставить, все равно, как бы, нейронка может что-то выдумать и выдумывать постоянно. А так, ускорение, соответственно, работы специалистов это факт, если, соответственно, внедрять какие-то инструменты уровня более высокого, чем чаты, это можно достичь в принципе на некоторых задачах, особенно рутинных, где в принципе современные нейронки показывают хорошие результаты, от 3 до 10 раз можно ускориться. Да, на серьезной, соответственно, логике, на серьезных проектах зачастую можно потратить столько же времени, а то и больше, на отлаживание того, что нейронка навдумывает, но об этом позже.

00:14:28

Мультимодальность, соответственно, сейчас современные модели прекрасно работают, как с текстом, так и с изображениями, так и с кодом, и все в 1 месте, можно скинуть, даже в принципе, визуальный баг скриншотом, и зная код, как бы, вполне, я проверял, может найти место, где это происходит. Снижение ошибок, за счет человеческого фактора, да, это будет чуть подробнее дальше, это скорее ближе к код ревью, то что человек в принципе, даже хороший специалист может иногда пропустить, машина как раз таки вот на банальных каких-то вещах, на банальных пропусках, там где-то что-то, забыли поменять переменную, она показывает себя прекрасно. Ну и как вишенка на торте, то есть, я все-таки считаю, что нейронка это не замена инженерам, не замена разработчикам, это возможность для них сфокусироваться на действительно сложных вопросах, уйдя от какой-то там рутины, переключивания json-щиков, написание банальных трудов и так далее, а заниматься проектированием, архитектурой, то есть то, чем пока как раз таки нейронка может быть либо слаба, либо очень рискованна. Касательно рисков, ну в принципе классика шамра, о чем все говорят, утечка данных, но да, данные могут утечь какие-то, но касательно программного кода, ну не знаю, мое мнение, если вы не работаете на оборонку или не пишите соответственно софт для космических кораблей, но антропикам или гуглу ваш код не сильно нужен, у них есть ресурсы, чтобы и без утечки в нейронку его повторить.

00:16:33

Треды, какие-то конфиденциальные данные, ну тут если они у вас находятся в репозитории, то у меня уже в принципе для вас плохая новость, у вас уже все небезопасно, Соответственно, если их там нет, то и утекать нечему, опять же все, даже при использовании каких-то агентов, все это настраивается, особенно если агенты с открытым исходным кодом, то это

можно еще и дополнительно все в программном коде настроить, поэтому тут важно просто-напросто аккуратно использовать как любой инструмент. Качество, непредсказуемость, потенциальной уязвимости. Факт, да, если соответственно дать нейронке напиши, сделай мне хорошо, и сказать, вот напиши мне CRM систему или еще что-то, ну соответственно будет и такое же качество, ну в принципе опять же, если мы возьмем первого попавшегося специалиста с фриланса и просто дадим ему эту задачу, результат может быть примерно тем Как улучшить качество, это я расскажу в последующих слайдах. В целом работать с этим можно, главное соблюдать осторожность и определенные практики. Сложность поддержки черный ящик в коде, тут на мой взгляд опять же тот же самый риск и есть и у людей, если за ними не следить, если не проводить код ревью, если не давать качественное ТЗ, будет ровно то же самое.

00:18:18

Если же передавать какие-то определенные задачи нейронки и за ней следить так же, как и за обычным человеком, учитывая, что есть подобный фактор, то это все можно перевести в контролируемость. Деграция навыков, атрофия экспертизы, ну, в целом, да, но мир к этому все равно, в принципе, идет, просто нужно, действительно, если только и делать, что пользоваться нейронкой, и копировать, ставить и вообще не думать, ну действительно мозг, как и в принципе мышцы, если ими не заниматься, он будет атрофироваться, он не будет думать, если же, опять же, использовать нейронку для какой-то рутины, ускорения и прочее, и наоборот, оставлять себе более сложные и серьезные задачи, то, я считаю, может пойти и наоборот. Да, я, кстати, здесь бы, наверное, даже дополнил, Серег, по этому слайду, Для меня, например, как для не разработчика оказалось наоборот с помощью еі я в кодинге чуть-чуть прокачался. То есть, какие-то вещи, которых я не знал, я их узнал и был вынужден им научиться, как разворачивать у себя Linux на компьютере с Windows. В общем, все, что связано с тем, чтобы начать кодить и какие-то термины по ходу, они вот благодаря тому, что я это начал, в этом смысле получилась не атрофия, а наоборот.

00:19:50

Возможно, для опытного разработчика, если он будет постоянно только кодиться и делать Ctrl+C, Ctrl+V, это будет действительно потеря навыка. В случае менее опытных, наверное, как-то по-другому. Или может быть такое, если мы рассматриваем сейчас особенно тех, кто начинающий разработчик Джунэ, если они не понимают каких-то глубоких деталей разработки, не начинают кодить с помощью ИИ, тогда они не понимают, в общем, что за код выдается им. То есть, им в этом смысле тяжело. И вот это тоже проблема, потому что сейчас женам, мне кажется, все сложнее будет войти в профессию разработки, потому что они не будут понимать то, что им машина пишет.

00:20:34

Как я, например, не понимаю, когда мне какой-то код выдается, я просто его применяю и смотрю работает он или нет. Но насколько он хороший, качественный или плохой, тут вот, условно, мне нужно идти к Сереге и спрашивать его, чтобы он трезвым взглядом на это посмотрел. Вот это, наверное, такие вот подводные камни, которые здесь могут быть. Я полностью с тобой согласен. Для джунов это, на самом деле, очень и очень опасная штука, очень велик риск соблазнов вообще не включать мозг, вообще не думать и пытаться всё, соответственно, просто-напросто копировать.

00:21:17

И, ну, сейчас, я бы даже сказал, есть тенденция, очень много таких людей приходит на собеседование, которые готовились непосредственно войти в IT, вот просто с нейронкой. Опять же, тут есть 2 подхода, то есть, можно пытаться нейронку, чтобы она объяснила, чтобы она работала тебе ментором. Сейчас, в принципе, люди и языки с помощью нейронок изучают и так далее. Но это должен быть соответствующий склад ума и желание развиваться, желание пытаться и не поддаваться соблазну копировать, вставить. Мне попадались на собеседованиях и разработчики, и аналитики, и кто только не попадался, кто именно.

00:21:58

Да, реально, то есть, 1 раз помню даже пытал 1 аналитика, но я смотрю, он не может толком ничего объяснить из того, что у него расписано в резюме. Что резюме за него написала нейронка, что готовился он 2 месяца просто-напросто. Прогоняя какие-то кейсы, задачки аналитики через нейронку, он даже не углублялся, то есть он просто решил, что, ага, ну вот я нашел список задач для аналитика, такой задачник, прогнал их все через нейронку, она прекрасно что-то ему описала на эти задачи, он даже не понял что, нарисовала какие-то схемки, диаграммки, он их вставил все в портфолио, и он на полном серьезе думал, что в принципе это прокатит и на работе, то есть он точно также будет все задачи кормить нейронки, и их потом вставлять. Да, да. И в этом, кстати, большой риск такой IT-компании, потому что иногда люди приходят, потом они что-то делают, но все это делается с помощью.

00:23:00

И, конечно, это все вскроется рано или поздно, но иногда, да, вот много кейсов в интернете тоже есть в открытом открытом Да, 1 раз пролетел кое-как через собеседование разработчик, да, и он, ну продержался он буквально 3 дня, на третий день его уволили, но 3 дня он старательно пытался лить код, вот просто скопированный из нейронки. Да, вот по поводу утечки данных, тоже еще такой момент интересный хочется посвятить, действительно, что часто вот сталкиваюсь с такими переживаниями на тему какой-то чувствительной информации, ну, то есть, если условно там какие-то ключи безопасности или там какие-то еще, в общем такие конфиденциальные

данные в коде не хранятся, а они не должны там храниться, то тогда здесь нет проблемы с тем, чтобы кодить с помощью ИИ или переживать за какую-то информацию, за ваш код, который будет где-то ИИ. Опять же, тут важно понимать, что ИИ не использует данные напрямую, как где-то у себя в хранилище. По сути, даже если ваши данные утекли до обучения модели, по сути, они просто легли в некий общий, не знаю, как бы это представить, такой огромное облако знаний, где все со всем связано, но это не воспроизвести потом полностью дословно. Условно, если вы загружаете какой-то файл, этот файл превращается в кусочки текста, эти кусочки текста отправляются в модель, но модель не запоминает их так, чтобы потом воспроизвести ровно то же, что вы ей скинули.

00:24:37

То есть, она не соберет с этого обратно файл. Поэтому здесь, наверное, тоже можно сильно не переживать. Вот, ну кажется, что плюсов пока больше, чем минусов, по крайней мере в том, чтобы пробовать это как-то внедрять. Серег, я тут немножко вклинился к тебе. Я даже по этому поводу могу добавить, но в принципе как бы риск утечки конфиденциальности, как бы, благодаря человеческому фактору, возможно, он даже выше.

00:25:07

В принципе, базы утекали, как бы, и у Яндекса, и у Мегафона, Соответственно, как бы, никто от этого не застрахован, человеческий фактор имеет место быть, и я на своей практике помню, я не раз ловил, соответственно, в коде, как бы, во время ревью какие-то, соответственно, данные, которых не должно быть в коде. При этом нейронка вполне себе способна проводить аудит огромных проектов, огромных массивов данных на безопасность. Более того, недавно наткнулся на интересный кейс, как раз таки, в рамках экспериментов на эту тему прогонял, так сказать, 1 проект. При этом, соответственно, до кейса самого дойдем дальше, но суть была в том, что, как бы, вот, близко к твоему кейсу, что ты рассказывал, ты написал, соответственно, как бы, вот, я также поручил тоже, как бы, написать, посмотреть, что получится. Потом, как бы, поручил, загрузил, как бы, в другую заново, в новый контекст, и код упал.

00:26:15

Я начал разбираться, что за ошибка, оказалось, что ему не понравился просто-напросто example файл, то есть, файл примера с кридами определенными, просто потому что там были слова, там, паспорт, там, и так далее, соответственно, как бы, и там были фейковые данные, но были данные, и у него сработала система безопасности, соответственно, то есть, я потом уже у них документ прочитал, то есть, у них есть да, то есть, как бы, они сами блокируют, если он находит какие-то такие данные, заставляют его упасть. Интересно, интересно, хорошо, можем двигаться дальше, наверное, да. Соответственно, здесь, ну давай немножко поговорим про инструменты, думаю про did research, ты им тоже активно пользуешься, тоже можешь что-

то добавить. Как по мне, это на самом деле очень-очень-очень большой шаг вперед, фактически это мультиагентная система с выходом в интернет и способная к длительному исследованию. Кейсов для ее использования, ну, на самом деле, огромное количество, и современные модели дают прекрасно, ну очень высокое качество результатов, при этом, ну гораздо зачастую все это получается объемней и, естественно, быстрее, чем если поручить такую задачу человеку.

00:27:52

То есть в целом я обычно туда закидываю какие-то задачи, которые в принципе я мог бы делегировать человеку, но как бы людей всегда не хватает, люди ограниченный ресурс, нужно ждать пока человек освободится от задач и так далее. Здесь я могу загрузить сразу несколько, соответственно, окон, несколько моделей, несколько нейронок своим вопросом. Оно может уйти на час, на 2 в исследование, думать, перерывать, соответственно, как бы, все интернеты, анализировать и готовить отличный, саморезованный результат. Тут может быть как, в принципе, какое-то исследование на тему, как применить ИИ, или как реализовать, там, допустим, интеграцию ИИ во что-то, то есть прекрасно находят кейсы крупных компаний, которые уже это внедряли, прекрасно находят различные способы. То есть вот сейчас дальше будем говорить про МЦП то есть технология уже приобрела популярность, так как выпустили ее Anthropic, OpenAI уже подхватила, внедрила к себе, Google после, но насколько я знаю, тоже уже внедрил, может быть еще не выпустил, еще не проверял на нем, но, по крайней мере, они подтвердили то, что они тоже ее будут поддерживать, то есть это такая общая универсальная технология, которая подходит, в принципе, к любым ИИ.

00:29:30

И уже готовых МЦП-серверов, их уже 1000, то есть технологии буквально полгода, то есть ее в конце прошлого года выпустили в свет, их уже 1000, как бы, что-то выбрать и так далее. На самом деле это можно потратить очень много времени. Банальный кейс, вот, я захотел, соответственно, как бы, интеграцию с ГИТЛАБ. Уже писать, как бы, самостоятельно этот МЦП-сервер, ну какой смысл, если есть уже готовые проверили на решение. Но их десятки.

00:30:04

Разбирать, соответственно, все по функционалу, все по возможностям, ну, это часы времени, время деньги, время дорого, времени никогда нет. Соответственно, через Dip Research описал задачи, которые мне необходимо выполнять, которых я жду. Соответственно, получил 2 десятка вариантов, и из них отобраны конкретно 3 варианта, которые подходят под условия моих задач, с плюсами, с минусами, с преимуществами, и даже с инструкцией по запуску. Поэтому, кто не пробовал depresearch, очень рекомендую. Да, наверное, тут я бы дополнил, я бы немножко знаешь, как это переводчиком с

технического до более такой понятный, может быть с нами кто-то есть, кто не очень хорошо понимает определенные вещи.

00:30:59

Вот deep research это, по сути, возможность задать какой-то вопрос на изучение модели и моделька пойдет шерстить интернет десятки и сотни ссылок источников разных, собирать из них информацию, перепроверять, то есть там есть условно можно представить это как целый аналитический отдел, который идет в интернет по вашей задаче. Допустим, вы хотите выбрать клавиатуру, как вот в кейсе, который Серега приводил, и вы сомневаетесь между разными вариантами, и вы можете задать вопрос: Какую клавиатуру выбрать для моей задачи? И описываете вашу задачу, например, что вам нужно, чтобы она была механическая или вам нужно, чтобы она была легкая, которую удобно взять дорогу. В общем, задаете какие-то параметры, которые вам нужны. Но это только 1 из примеров.

00:31:53

Ну и вот в этом случае, фактически, есть модель, которая выступает как главный аналитик, назовем это так, который раздает задачи дальше младшим аналитикам, которые идут каждый по какой-то своей тропинке или в свой уголок интернета и там они проводят исследование, возвращаются с результатами к главному аналитику, но сначала приходят еще к некому судье, так называемому, который валидирует эти данные и, условно, принимает результат, как подходящий или не принимает, просит его скорректировать. Но, если простым языком, то, фактически, это вот целый аналитический отдел в аренду для разного рода задач. То есть, начиная от того, какие-то вопросы, связанные с, ну вот я, например, при переезде некоторые вопросы изучал с помощью deep research и там пытался найти информацию об определенных локальных нормативных актах, законодательстве или чем-то еще. В этом случае Deep Research еще чем хорош? Он выдает ссылки на источники, с которых он взял эту информацию, и эти ссылки, вы можете потом по ним перейти и провести факт чекинг, действительно ли то, что выдала нейронка, соответствует правде.

00:33:12

Тут стоит сказать, что никаких гарантий того, что правдивая информация будет в интернете, даже если вы просто пошли туда и нашли ссылку, тоже, конечно, нет. Ну, то есть, есть какая-то вероятность, что нейронка ошибется, потому что кто-то, кто писал какую-то информацию в интернете, заранее тоже ошибся, потому что ну нейросеть, в этом как раз ее сила, она собирает большое количество источников информации, то есть, там последний раз, когда я делал какой-то запрос, было там 300 с лишним ссылок она промониторила и выдала мне сагрегированную информацию. Думала, наверное, минут 30. Вот она 30 минут ходила по интернету, анализировала и вот вернулась ко мне с результатом исследования, вот. Запускается он на

практике через, то есть, есть оригинальные интерфейсы, вот этот функционал есть в Клоде от компании Антропик.

00:34:03

Из России он немного тяжело запускается, скажем так, но возможность его запустить есть. Есть в оригинальном чате GPT, тоже есть функция глубокое исследование и вот в таком случае моделька идет вот как раз по такому пути, как я описал. И есть в perplexity это отдельный такой поисковик, в который тоже встроен depresearch, очень хорошо работающий. И вот интересно, что, например, в perplexity он менее цензурирован. Если вы пойдете в чат gpt или в kloda, он очень сильно заточен на определенные там политкорректности, некоторые другие аспекты, а вот perplexity он, ну, скажем так, более прямолинеен в том, что он просто собрал информацию, соответственно, как собрал, так и донес.

00:34:53

Мы в alpina gpt тоже добавим этот инструмент в какой-то обозримой перспективе, планируем у нас это в 1 из приоритетов находится, Поэтому в какое-то ближайшее время, там, я думаю, в течение лета, появится, наверное, и в Alpina gpt. Тогда вы сможете воспользоваться этим функционалом у нас. Попытался описать коротенько, так, более простым языком. А то, что касается MSP, наверное, Серег, тебя попрошу перевести это немножко, что же это за штука такая человеческим языком. Хорошо.

00:35:35

Чуть-чуть добавлю к Dip Research, какое еще очень простое и хорошее применение, соответственно, как бы его. В принципе, тут как дипресировщик ODA, так и OpenAI, так и perplexity подходит, в принципе, мне больше всего нравится perplexity, потому что даже без супер углубленного режима, то есть он прекрасно, как бы, шерстит, все это полезно может быть в первую очередь для разработчиков кейс, то есть, когда есть какая-то проблема с каким-то пакетом или еще чем-то, вместо часов прерывания с TechOverflow или Eachue на гитхабе, можно это решить за несколько минут, вот банально, то есть, только на днях был кейс, у нас используется система документации Афина, это стартап, то есть у них периодически бывают какие-то проблемы, вот, ее обновили, у нас все, соответственно, заблокировалось, ограничилось, хотя они обещают безлимитный доступ на своем сервере. В прошлый раз, когда полгода назад была эта проблема, я потратил вручную часа 4, прерывая собственно сотни их ишлю на гитхабе, о том, как это решить, где в итоге, да, они в своих ишлю, там сами подсказали решение, но их была тьма, и тьма способов. В этот раз я использовал perplexity, задал вопрос и получил решение через несколько минут, соответственно, что вот у них изменилась эта система, соответственно, так и так, и теперь решение такое-то, оказалось, мы там в другую таблицу перенесли эти настройки, То есть, экономия несколько минут против 4 часов.

00:37:23

Ну и касательно МЦП. Эта вещь позволяет, в принципе, внедрить нейросеть внутрь бизнеса, внутрь каких-то, в жизнь фактически. То есть, как бы, нейросеть, как бы, когда она отделена от контекста, она, ну, как по мне, убогая. Да, как бы, она может давать прекрасные ответы, но отделенная, соответственно, как бы, от мира, она не несет такой пользы. Какие примеры могут быть?

00:37:57

Опять же, ну, сейчас в любой компании стандарт это использование Git. Есть, соответственно, Gitlab, есть GitHub, где хранится код. Наверное, тогда можно тоже, я поработаю переводчиком, То есть, есть некое место в интернете, куда все разработчики компании заливают свой код. И вот там, соответственно, этот код хранится и все новые какие-то доработки, которые возникают, они туда подливаются, скажем так, да. То есть, постепенно кодовая база наращивается, но хранится она не на локальном компьютере, условно, у разработчика 1, у разработчика 2, а чтобы проект работал, оно все живет в облаке.

00:38:42

Вот таким образом GitHub, GitLab являются такими платформами, где живет вся кодовая база. Да, соответственно МТП позволяет интегрировать нейросеть, как в принципе, как я сказал, как USB, то есть, позволяет соответственно нейросети получать данные из любой системы и их использовать, ну и в обратную сторону тоже. То есть, банально пример, вот у меня даже есть здесь заготовленный, то есть, есть task трекары, где ставятся задачи, соответственно, есть Gitlab, где хранится код. Соответственно, используя 2 таких MCP сервера, 1, соответственно, для интеграции с Task Tracker, другой с Gitlab, можно скормить нейросети какой-то merge request, то есть, какие-то изменения в коде на проверку, она может дать детальные, соответственно, комментарии по этому коду, причем, ну, зачастую, то есть, я сам проводил код-ревью не раз, но в 90% случаев, вот настолько детально, как бы, описывать у меня нет времени. У нее есть, и занимает это не часы, а минуты.

00:40:10

Соответственно, также благодаря этому МЦП-серверу она может зайти, соответственно, в трекер, ознакомиться с задачами, проверить, соответственно, соответствие кода, как бы, что эта задача была в ней реализована, откомментировать, то есть, этот комментарий непосредственно оставила нейросеть уже внутри. То есть, для разработчиков, я думаю, в принципе, эта история понятная, для не разработчиков это может быть любая CRM-система, тот же самый Битрикс, АМА CRM, все, что угодно, то есть, все, к чему есть какое-то API, можно подвязать под нейросеть, и она может, соответственно, как получать оттуда данные, так их туда и передавать, она может их анализировать, она может, соответственно, их обрабатывать, выдавать какое-то саммари, давать какие-

то комментарии. То есть фактически это звучит так: мы, грубо говоря, открываем нейросети дверь к какому-то нашему хранилищу с нашим кодом, с нашими материалами, базами данных или чем бы то ни было другим и нейросеть постоянно имеет доступ. То есть нам не нужно каждый раз дверь открывать, она всегда для нее открыта и тогда нейросеть может работать с этой информацией на постоянной основе. Я поглядывая за таймингами, у нас иногда немножко модерировать, наверное можем двинуться дальше и как это, пойти к следующему блоку.

00:41:44

Ну тут давай тогда возьми, ты слово расскажи про свой пример. Да, ну собственно мой пример, я его словами опишу и возможно там в какой-то момент потом с тобой, Серег, там экран другой зашарим, вот. Но, если очень кратко, у меня была такая прям конкретная задача, моя личная боль, очень большое коммуникации в телеграмме, в разных чатах, папках и не всегда. То есть, иногда для того, чтобы понять и определить, что что-то важно, нужно прочитать все чаты во всех папках. Вот, и только после этого ты понимаешь, в первую очередь нужно ответить, на что во вторую и возникало много с этим проблем.

00:42:26

И когда у меня наболело совсем, я решил попробовать решить свою задачу с помощью нейросети. Сначала я пошел просто в интерфейс Клода и попробовал в нем заходить какое-то простое решение, создал отдельный проект и в этом проекте писал код, который дальше вставлял. Опять же, мне пришлось разобраться, как установить себе python на мой компьютер, как там развернуть все эти вещи. Я это все делал абсолютно полностью с нейросетью. Вот прям до этого я никогда не писал код и, как бы, только, то есть, я понимаю теорию какую-то, да, но вот прямо руками что-то писать, я не знаю, языка, не знаю, синтаксиса и многих вот этих вещей.

00:43:10

Поэтому для меня это был такой тоже вызов, как вообще разобраться в этом. Собственно, нейронка меня за руку провела, прямо, то есть, я прямо ей сказал, говорю, вот для чайников вообще в этой части объясни вот пошагово все, что нужно делать, вплоть до того, как установить и так далее. И попутно я скидывал скриншоты, когда я что-то делал, я прям скидывал скриншот, вот смотри, что получилось. Она так все, молодец, ты двигаешься по плану. Следом подсвечивала мне куда мне дальше нужно идти и что дальше нужно установить.

00:43:43

И последовательно, пошагово я пришел к тому, что вот у меня установился VSL код, специальная программа, в которой я могу там какой-то код писать. И дальше я попробовал в ней создать, собственно, этот проект. То есть, я вместе с, я писал свою задачу, прямо вот конкретный кейс, я прямо так и сказал, вот моя проблема слишком много коммуникации, мне нужно чтобы

был бот или какой-то другой способ мониторить мой telegram, выделять все сообщения, анализировать их, анализировать их контекст и дальше выдавать приоритизацию, на какое сообщение обратить внимание в первую очередь, на какое второе и краткую выжимку саммари по итогам дня. И сначала, естественно, просто искал в интернете такой сервис, то есть мой первый путь был, я попытался найти какое-то готовое решение. Готовых таких решений я не нашел, с удивлением для себя, потому что я думаю, ну наверное такая проблема не только у меня существует, но готовых решений не было и вот тогда меня это побудило собственно создать какое-то свое.

00:44:48

У меня здесь тоже была развилка. Первое пойти в разработку и договориться, не знаю, с кем-то, с разработчиком, чтобы он для меня что-то закодил. Но в таком случае, все равно так немного чувствительная область, то есть, мне нужно выдать свой личный телеграмм, там у меня какие-то данные могут храниться чувствительные и так далее, и так далее, а мне нужно, получается, дать к нему доступ. И вот в отношении ИИ у меня почему-то было меньше переживаний, что, ну, как бы, иишки на самом деле мои чувствительные данные, логин и пароли от telegram не нужны. В итоге получилось некое решение, программу и код написала ИИ, я дальше развернул, и у меня начались, как бы в отдельном телеграмм канале специальном таком, персональном, мне начали сыпаться уведомления о том, что вот мне написали, приоритетность.

00:45:41

Потом, дальше, я начал развивать этот сервис, просил его докручивать, например, делать какую-то заготовку ответов на некоторые сообщения и с удивлением обнаруживал все больше и больше, что в общем проект дорабатывается, разрастается. В какой-то момент он разросся до того, что пришлось его разбивать на так называемые микросервисы, как в разработке это называется, когда есть сервис отвечающий за 1 часть программы, работая за другую и так далее. И постепенно я пришел к тому, что установил себе отдельный специальный, у клада есть такой сервис, который называется Cloudcode, который позволяет развернуть у себя на компьютере нейронку, которая будет иметь доступ к вашим папкам, иметь доступ к вашему проекту целиком и как раз вот то, о чем Так, жимал, ты кажется завис. Так, ну, пока Жемала нет, тогда я немного продолжу. Соответственно, то, что Жемал рассказывает, это как раз-таки то, что сейчас называют активно VIPЕ-кодингом.

00:47:19

Могу даже показать, в принципе, как выглядит этот Cloud-код, это 1 из инструментов, который, в принципе, я бы сказал 1 из лучших, то есть OpenAI есть кодекс, и он даже опенсорсный с открытым кодом, но в сравнении с Cloudcode Антропиков он несколько лучше выглядит. Это вот так, это такая консольная штука, и запускается, ей можно на человеческом языке давать

какие-то команды, соответственно, здесь я вот для примера просто-напросто взял проект и расскажи, как работает авторизация на этом проекте, и он пошел изучать все его файлы, соответственно, и описывать, что как работает, то есть на самом деле это очень полезный интересный кейс, когда прилетает какой-то проект с неизвестным кодом, можно с помощью него быстро изучить, разобраться, то есть вот даже, в принципе, время жизни токена подсвечивает, что и как, он, в принципе, на самом деле, дает очень хорошие, очень полезные, соответственно, ответы на неизвестной кодовой базе. Дополнительно он может, в принципе, как же мало рассказывал, и программировать самостоятельно. Да-да-да, в принципе, может и рефакторить, но тут, я бы сказал, очень такая опасная штука, если ему скормить, вот, из разряда: Вот тебе проект, сделай его из нехорошего хорошим, он может сделать еще более плохим. То есть на слишком широком контексте он всё равно захлебывается.

00:49:19

Да, под присмотром, плюс с очень большим сужением контекста. Чем хуже задача, которая ему дается, то есть, вот, пролифактор такой-то модуль, соответственно, с такой-то задачей, тем он лучше справится. Если, соответственно, он пойдет молотить, рефакторить всё, это прям провальная история будет, он сожжет пару сотен долларов в никуда, наломает весь проект, потом еще, в принципе, сам напишет автотесты, потом, чтобы эти автотесты запускались он переписет под них код, как бы, лишь бы они запускались, то есть, тупиковая задача. Если же, соответственно, как бы, это какой-нибудь небольшой микросервис или какой-нибудь модуль, он может прекрасно с этим справиться. Так, Gimal вернулся.

00:50:19

Да, видимо. Показал, как выглядит Cloud-код. Нейросети, видимо, решили, что не нужно рассказывать про секреты вайп-кодинга и меня выкинуло. Да, вот как раз я остановился на том, что в итоге с Cloud кодом я как раз его поставил к себе на ПК и вот, по сути, теперь я полностью работаю со всем контекстным проектом и продолжаю его разрабатывать. Для меня это, конечно, очень интересный опыт был, который позволил мне. Возможно, моя задача не такая сложная, чтобы все-таки ИИшка ее кодила напрямую, да и вот те риски, о которых говорил Сергей, что, возможно, мой код не очень качественный или, возможно, не очень безопасный, я этого не знаю и не могу проверить.

00:51:09

То есть, для этого мне нужно было бы отдавать его какому-то техническому специалисту на оценку, но мне самое главное, как конечному пользователю, что все работает. В общем-то я хотел бы, чтобы работало и оно работает. А то, как оно было сделано, наверное, в этом плане даже для меня было вторично. Поэтому для тех, кто является разработчиком, у вас, например, есть какая-то идея. Опять же, перекладываем на прикладные рельсы.

00:51:35

Где это хорошо применимо и могло бы быть вам полезно? Вы хотите сделать какой-то прототип, протестировать гипотезу, какую-то минимальную версию продукта сделать, возможно, вам нужен какой-то лендинг-пейдж наподобие того, который вот сегодня Сергей показывает в качестве презентации. Это все можно сделать с помощью кода. Это можно делать или через курсор, есть отдельный такой сервис, в котором тоже можно писать код, или в клауд-коде. В каждом из них можно плюс-минус вот такие задачки решать, и они довольно хорошо там решаются.

00:52:19

Но, конечно, большой сложный такой проект, который используется в какой-то коммерческой среде. Тут я абсолютно поддерживаю мнение Сереги, что не стоит полностью доверять ИИ, бездумно копипастить код или просто запустить его в проект и сказать: Вот теперь сделай хорошо и полностью довериться тому коду, который был написан. Мой кейс примерно такой. Наверное, если у нас останется время ближе к концу, то, может быть, я покажу, как я его делал. Полезные ссылки на курсор, Cloud-код, инструменты depeserch и так далее поскидываем тогда.

00:53:00

Ты остановился, Сереж, на том, что ты уже показал Cloud-код? Да, я показал, как выглядит его консолька, и даже пример из разряда расскажи, как работает авторизация на этом проекте, где он выдал очень детальное описание, даже с имеющимися ролями. Супер, я в конце, наверное, покажу свою версию, как у меня в WSL это работает. Можем тогда двинуться дальше по твоим слайдам? Да, что я бы здесь хотел сказать.

00:53:34

Когда двинуться дальше по твоим слайдам? Да, в принципе, тот подход, который ты рассказал, он имеет. Что я бы здесь хотел сказать, да, имеет место быть, почему бы и нет, Но, в принципе, тот подход, который ты рассказал, он всегда можно сделать чуть лучше, имеет место быть, почему бы и нет, но всегда можно сделать чуть лучше. То есть, я ради интереса пробовал, как раз таки, и вот так вот, да, вот, бахни, мне вот такое вот, огромное, сделай хорошо. И даже периодически, вот, по мере выхода моделей новых, я закидываю специально в них какую-нибудь задачу из разряда спроектируй, там, и напиши 3 микросервиса с такими интеграциями и так далее, на которые, в принципе, как бы, многие мидлы загнутся, соответственно, как бы, и вот мне интересно, на каком этапе оно захлебнется.

00:54:50

Пока что, как бы, захлебываются, в спойлерах, да. Начинают в какой-то момент выдавать, не сосвоятся, как бы, и уже видно, что дебащить потом вот это дело выйдет дольше, чем описать самому. Соответственно, как бы, но ситуация улучшается, то есть, как бы, я это пробовал в версиях кодов, вот, на 3, на 35, на 37, на 4 вышел, и с каждой версией он на самом деле всё

ближе, всё сильнее, как бы, у него окно возможностей, соответственно. Но как можно это в принципе улучшить? Пока, как бы, не прошли десятилетия, как бы, и нейронка всё-таки не научилась сама писать качественные продукты, как можно улучшить, соответственно, это разделение ролей, то есть не кормить нейронки, сразу вот тебе задача, делай ее, радуйся.

00:55:53

Берем процесс работы, как это работает в IT-компаниях, то есть есть определенные роли, есть архитектор, есть аналитик, есть проджект менеджер, есть разработчики, есть тестировщики, и задача ходит по этим ролям, то есть, как бы, это старый, объезженный на человеках процесс, и к нейроне он оказался вполне себе тоже применим, когда у тебя есть агент, соответственно, ну фактически промт, то есть даже сейчас мы не будем, это, наверное, уходит далеко за пределы данного вебинара, уходить в ИИ агентов, соответственно, какие-то цепочки мультиагентные и т. Д. Это может быть просто-напросто несколько промптов заготовленных, так сказать, системных для ассистентов, с ролью аналитика, с ролью, соответственно, архитектора и так далее, и ты кормишь свою задачу сначала промпту-аналитика, при этом желательно, чтобы вот тоже рекомендация, промт должен быть действительно большой и качественно описывающий, не из разряда ты аналитик, ты хорошо анализируешь. С детальными практиками, то есть, ну вот я экспериментировал, то есть, опять же, сам промт мне помог написать сам код, то есть, как бы, я ему описал, чего я хочу от этого промпта, он не, ну совместными итерациями правок, дополнений, то есть, как бы, на несколько листов А4, на Гигнегелях, или, соответственно, как бы, этих промптов, инструкций и так далее. И, соответственно, есть этот промт-аналитик, то есть, как бы, ему скажешь задачу, он уже пишет детальные бизнес-требования, юзер-кейсы и прочее.

00:57:32

Чтобы, соответственно, как бы уменьшить свои временные затраты, можно загнать это в другого, соответственно, в другой. Главное разный контекст, так сами Антропика рекомендует, то есть не делать все в 1 контексте, то есть не писать этому же ассистенту. Проверь сам себя. Он начнет, соответственно, подгонять результаты своей проверки под свой ответ. У них тоже есть какое-то эго.

00:57:58

А загружать его результаты в отдельный контекст, соответственно, там проведи ревью, проверь, дай ошибки. Выдает ошибки, загружаешь их обратно туда, он их исправляет, и так туда-обратно. Ну а финальное ревью, естественно, как бы ты заказчик, ты уже сам принимаешь, как бы, и говоришь, что там еще поправить. Дальше, соответственно, передаешь это, грубо говоря, агенту системному аналитику, спроектируй, соответственно, опиши ТЗ для разработки и прочее. Такая же цепочка между, так сказать, агентом-аналитиком, агентом архитектором, ревью-правки, ревью-правки,

соответственно, дальше, ну, если есть компетенция, естественно, проводишь финальное ревью сам.

00:58:52

Если нет, в любом случае результат от вот такого процесса будет выше, чем без него, даже если ты сам не имеешь компетенции проверить соответственно спроектированную этим и агентом архитектуру. Но, вот, с моего эксперимента, на первом же этапе ревью, агент-ревьюер назвал проектирование агента, который его сделал, технологическим фетишизмом, дословно. Ира вписал 2 листа правок о том, что бизнесу это не нужно, там действительно был пятилистовый промт о том, что он должен не только в технологии смотреть, но и в бизнес, в реализацию, соответственно, не городить, соответственно, как бы, что-то огромное и ненужное, и так далее. По итогу получилась, соответственно, как бы, минимальная для MVP схема, то есть, таким образом я как раз таки экспериментировал с МЦП агентом, которого еще не существует, вот, для Афины, чтобы ее интегрировать, для нее нет готового, поэтому, ну вот, таким образом я решил проэкспериментировать и написать его. После, соответственно, когда у тебя уже есть спроектированная архитектура, вот этими, соответственно, предыдущими агентами, ТЗ для разработки, это всё можно задать следующему агенту, который мог дать задачу.

01:00:12

Разбей это всё на малые задачи длительностью, там, 1-3 часа, соответственно, со строгим ожидаемым результатом и требованиями. Он разбивает это на 15-20-30 задач, у каждой есть требование, соответственно, и вот эти задачи ты уже кормишь тому, вот, своему агенту, тому же самому клауд-коду, и говоришь ему: Двигайся точно по задачам, не пропуская, не меняя последовательность, разрабатываешь по задачам, каждую задачу перепроверяешь, покрываешь тестами, и только после полной проверки этой задачи переходишь к следующей. И вот при таком подходе, я тебе скажу, процесс получается гораздо более качественный и надежный, то есть, как бы, и в принципе уходит, так называемый, вайп-дебаггинг. То есть когда крадываешься вайп-кодингу, как все классно, а потом 2 дня плачешь над вайп-дебаггингом, соответственно, пока как бы пытаешься заставить это дело работать. Да, с этим я тоже сталкивался в процессе.

01:01:29

Да, будем, чтобы успеть в тайминге, как раз у нас уже почти час прошел от вебинара, чтобы мы еще успели поотвечать на вопросы, которые наверняка тоже будут сегодня. Много слов таких необычных. И, наверное, что важно, хотелось бы показать для тех, кто не очень глубоко в теме, как вообще собственно начать этот процесс кодинга с помощью ИИ. Вот это мы тоже сейчас подетальной раскроем, когда расскажем весь материал. Еще на практике прямо покажем, зайдём в платформы, покажем, как это примерно делается.

01:02:07

Все, понял, тогда ускоряюсь максимально. Здесь сильно задерживаться не будем, то есть, просто скажу, что ну, помимо кодинга можно использовать и в аналитике, соответственно, в принципе процесс выше я описал, как можно гонять задачу по ревью, то есть, в принципе, можно проверять, что упущено, не упущено. Тут я еще хочу подсветить, опять же, кейс из разряда того, что я говорил, где не стоит или стоит доверять. То есть, как я уже рассказал, то есть, получилось интегрировать и с трекером, вот, и попросил я его, значит, проанализировать задачи сотрудника, и его, соответственно, эффективность. Он мне проанализировал, вот, выглядит очень круто, На самом деле, вот, он даже предложил повысить, соответственно, этого сотрудника, как бы, до тимлида и прочее, то есть, нет, на самом деле сотрудник, Кристина, молодец, я тут совсем полностью согласен, он даже подсвечивает, что нужно предотвратить ее выгорание, и так далее, то есть у нее действительно очень большая нагрузка и прочее.

01:03:18

Кристина с нами в чате пишет. Выводы он сделал отличные, но если детально углубиться, соответственно, как бы, в основание, то, как бы, половина задач, которые он анализировал, оказалась фантазией, и доверять этому без проверки нельзя, абсолютно, потому что он может сказать о тесте: Да вообще увольте этого сотрудника ни за что, просто на основе, соответственно, какого-то комментария в трекере и так далее. То есть, тут все нужно проверять, дотачивать. Я верю, что это можно дотащить, и он сможет анализировать все объемы данных действительно хорошо, но не рекомендую, соответственно, сразу доверять и отдавать это, я не знаю, дальше не проверяя. Ну, в принципе, здесь, я думаю, мое мнение ясно, что коммерческая разработка полностью в ии, ну, однозначно нет, то есть, как бы, помощь, сокращение затрат, да.

01:04:24

В случае, соответственно, интеграции, как раз-таки, агентов, ролей, МЦП и так далее, это может очень сильно ускорить, улучшить и удешевить разработку для бизнеса. То есть, за этим, я считаю, будущее, но это все нужно, опять же, пока что проводить через эксперименты, соответственно, как бы, через исследования, как бы, и внедрять аккуратно и постепенно. В педпроекте, которые, вот как ты сказал, пожалуйста, как бы, уже сейчас кто угодно может это делать, в принципе, как бы, ну вот, даже я могу, в принципе, как бы, прекрасно программировать на нескольких языках сам, но не всегда есть на это время. Так я повесил тот же самый клауд-код на отдельном мониторе, и вот, по той схеме, о которой я рассказал, то есть, как бы, МЦП агент для Афины, который сейчас мы внедряем, он полностью написан им. То есть, как бы, я занимался своими задачами периодически, просто давая ему какие-то правки, как своему сотруднику, как разработчику.

01:05:34

Но это внутренний проект, который не несет потенциальных рисков для бизнеса. Поэтому, ну, заменить разработчика, нет, заменить вебмастера, да, заменить, соответственно, как бы, фрилансера для каких-то, ну, не критичных проектов, там, лендинги, соответственно, какие-нибудь сайт, визитки и так далее, то, что, если это рухнет, как бы, это не принесет, соответственно, как бы, за день простоя убытки бизнеса, да, пожалуйста. Соответственно, как бы, если же, соответственно, это рухнет, и потом ИИ не сможет тебе починить, а бизнес теряет деньги, и тебе нужно искать разработчиков, соответственно, чтобы это починили, а они потом повертят, вот так вот, мы в это лезть не будем, тогда ну ни в коем случае, как бы, нет. Только для разработчиков, причем уровень специалиста всё равно должен быть, даже если он использует ИИ, задача должна быть та, которую он может выполнить, и выполнить хорошо, и может оценить её качество. Нельзя поручить джуну сеньорскую задачу, в надежде, что он сможет просто контролировать ИИ, и ИИ за него это сделает.

01:06:53

Это путь, очень опасный путь в пропасть, я бы сказал. Ну, в завершение, в принципе, кратко могу рассказать, вот то, где я вижу, что где можно явно уже применять в разработке, то есть то, где минимальный контекст, то есть, соответственно, не напиши мне весь проект, а вот прекрасно подходит микросервисная архитектура для этого, То есть, когда проект разбит уже на, человека желательно, грамотно на разные микросервисы, соответственно, внутри 1 микросервиса, который очень ограничен своей ролью и прекрасно может работать. Если это фронтенд, то есть FSD архитектуры, блок-паттерны, да, сейчас говорю для не разработчиков незнакомыми словами, но, грубо говоря, это что-то позволяющее сузить максимально контекст. Автотесты, отладка, то есть, сейчас прекрасно запускает доккер-контейнеры, соответственно, запускает автотесты, как бы, сам может их писать. Code review я показывал, соответственно, честно, его результаты и качества меня даже самого несколько удивили, я даже ожидал несколько худших результатов, когда начинал этот эксперимент.

01:08:09

Тестирование требований и кейсов, тоже, соответственно, опять же, но тоже вместе с аналитиками, вместе с тестировщиком, если это коммерческий продакшн продукт, то есть, как бы, в комплексе, то есть, соответственно, он может подсветить какие-то упущенные вещи, но нельзя это передать ему. Какие-то рутинные задачи, то есть, как я уже говорил, там, парсинг, соответственно, там, генерация документации и прочее, то есть, да, прекрасно. Опять же, на самом деле, какие-то вот такие вещи, мне нужно вот спарсить сайт разовые, да пожалуйста, как бы, скрипт запустил, как бы, пусть да, ладно, этот сайт обвалит, бывает такое. Но запустил, получил результаты, выкинул, как бы, и не надо никому платить. Ну, research анализ и различные MVP, соответственно, как бы, при этом, как бы, MVP могут

получиться при правильном подходе, гораздо более задокументированные и готовые к дальнейшему, соответственно, развитию, да, пожалуйста.

01:09:17 Но помним, что все равно важен контроль. В принципе, мы и так перелимитились, поэтому я, наверное, на этом закончу, выжимал дальше слово тебе. Да, спасибо большое, спасибо, Серёг, очень интересно и Especially, наверное, для тех, кто уже хотя бы чуть-чуть в теме. Я сейчас попробую немножко подсветить для тех, кто вот прям совсем не в теме, на очень каком-то вот прям банальном базовом примере, как это работает. Серег, я тогда перехвачу экран, наверное, а я сейчас покажу, как это работает.

01:09:58 Просто, как машина вообще в принципе пишет код и что это вообще такое. Вот, давайте. Все, включилась моя. Да, я сейчас вот включу. Это оригинальный интерфейс Клода, вот, и я сейчас пишу ему вот такую очень, ну, как сказать, топорно пишу prompt: напиши код для мини-игры, где нужно отбивать шарик из 1 стороны стола в другую.

01:10:26 Вот, и давайте просто посмотрим, выберем что-то с модель X4 orus и посмотрим, что сделает кот. Сергей, видимо, выключил просто камеру. Я вижу, что он с нами. И вот смотрите. Сейчас моделька думает, она поняла, что нужно сделать.

01:10:50 Я создам это как html-артефакт с полной функциональностью. Он делает игру pong и пишет код, как это будет работать. То есть, вот управление клавиатурой, что-то еще пишет. В общем, сейчас он создает код, который мы с вами сейчас проверим, да. Понятно, что то, что я сейчас делаю это такая совсем простая задачка, но, по сути, это тоже код.

01:11:18 И как это может работать дальше? Тут были вообще тоже вопросы. Смотрите, что получилось. Новая игра. Я создал для вас классическую игру Pong.

01:11:30 Вот основные особенности, геймплей: 2 игрока управляют ракетками, мяч отскакивает. Игрок 1 левая ракетка WS, игрок 2 стрелочки вверх-вниз. Давайте попробуем. Так, новая игра. Так, сейчас, сейчас, сейчас.

01:11:50 Видимо, надо Ага, вот, это работает, а WS почему-то не хочет. Так, так, так. Отбивание здесь уже происходит. Но явно тут можно еще улучшить. Используют Canvas для отрисовки.

01:12:13 Давайте попробуем еще раз. Может быть, он заработает. WS не работает. А вот прямо мы ему и скажем: не работает V и S. Хороший, кстати, вопрос.

- 01:12:36** Может быть, с раскладкой было дело, но нет. А, вот, кстати. Все, размышляю о причинах неработоспособности. Давайте исправим, обновлю код. Сейчас он обновит его.
- 01:12:53** То есть он правит. Вот версия 2. Смотрим. Ага, вот, все заработало. Так, оба.
- 01:13:07** Ну, в общем, вы поняли, как это работает. Понятное дело, что это был очень такой топорный пример, и я показал, как это может выглядеть, но, в принципе, на основе этого примера дальше можно моделировать что-то свое. Можно было бы попросить у него Landing Page. Он бы сделал html-код, который бы выглядел как страничка Landing, который дальше можно было бы сверстать на Тильде. Например, можно было бы закинуть ему какую-то задачу.
- 01:13:38** А теперь, а теперь, теперь сделай html код для лендинга, который будет продавать эту игру. Тоже топорный промпт. Я сейчас просто хочу показать вам, как это могло бы работать, не генеря что-то очень большое. И как бы выглядел лендинг, если бы его делал. То есть, чем больше вы контекста зададите, тем больше задач вы зададите, как бы больше примеров каких-то, возможно, дадите ему, тем более ожидаемо качественный результат вы получите.
- 01:14:22** Но сама суть, она, примерно, в том, что сейчас происходит на экране. То есть, как вы приходите в студию разработки, например, если бы вы делали лендинг пейдж, вы говорите, что вы хотите? Вы хотите какую-то страничку. Что она должна делать? Чем больше вы зададите инфы, текстов, каких-то деталей, тем более качественный результат вы получите.
- 01:14:43** Но, если вот так напишите, то в любой студии разработки вы тоже, возможно, получите, а не то, что вам бы хотелось получить. Сейчас я покажу вам, как работает deer research. Сейчас он закончит писать код. Мы ждем, пока он это делает. Так, пока он это делает, давайте, наверное, я еще переключусь быстренько, покажу вам, как работает.
- 01:15:10** То есть, то, что я показываю сейчас происходит в оригинальном интерфейсе Клода, то есть, у него есть свой интерфейс, десктопное приложение, или можно делать это на web-версии, и, таким образом, получать вот такой результат, как мы сейчас видим. Здесь также есть в настройках возможность включить тот самый dep research. У perplexity тоже есть такая же функция, соответственно, если мы, например, выбираем perplexity, давайте я перешерю, пока там код пишется, покажу вам perplexity. Вот, соответственно, perplexity. В perplexity тоже есть функция dip research.

- 01:15:49** Вот глубокое исследование, соответственно, можно в бесплатном режиме тоже 3 запроса в день, и вы можете что-то его спросить. Ну, например, я не знаю, какой-то какой-то вопрос, который вас интересует. Вот, кстати, давайте такой небольшой интерактив: напишите в чатик, какой вопрос вам бы хотелось исследовать, и мы сейчас посмотрим, что это replexity сделает. А я вам покажу лендинг, который получился вот прямо сейчас. Это, конечно, мне кажется, очень круто.
- 01:16:20** Смотрите, что получилось, что сделал Клод. Вот, смотрите лендинг. Ну, смотрите, с анимациями, со всякими тут, вон тут эти шарики летают по экрану сами, со всякими выезжающими штуками, смотрите, при наведении все рамочки подсвечиваются. Серег, ну это прям почти Alpina GPC получился, только для понга. Ну смотрите, вот демоверсия игры.
- 01:16:54** Тут, наверное, можно ее. Нажмите на экран, используйте VAS. Ну-ка сейчас а но демо версия пока не работает надо им потом сказать что говорят игроки отзывы написал смотрите со звездочками Невероятно затягивает. Начал играть в обед, опомнился только вечером. Классика, которая никогда не устареет.
- 01:17:21** Все, я пошел покупать за 299. Прямо сейчас есть возможность купить. Только модуль оплаты осталось прикрутить, дать ему ссылки. Безопасная оплата, 30-дневная гарантия возврата. При наведении смотрите все эти анимации выезжающие.
- 01:17:38** Даже в конце копирайт Pong Ultimate все права защищены. Созданы с любовью для любителей пластических игр. Ну вот, примерно про это я и говорил. То есть, раньше, чтобы сделать такой лендинг, вы понимаете, да, вам нужно было идти в какой-то конструктор, что-то там собирать. Вот для такого ленда, мне кажется, нужно было реально нанимать дизайнера и какого-то html верстальщика, типа, просто в конструкторе такое не соберешь.
- 01:18:05** Это нужно реально писать код. Вот здесь это сколько мне понадобилось? Ну вот, он пару минут, наверное, писал код. Это вот пример того, как можно работать с кодом, если у вас есть какая-то идея или вы хотите что-то протестировать, или вам нужен простенький лендинг пейдж, то дальше вот то, что вы получили здесь, вы же получили код, этот код можно скопировать. Вы можете вставить его, например, там в Тильду или куда то еще и, ну, собственно, он будет работать.
- 01:18:34** Микрофон, кстати, не слышно, у тебя выключен, Серег. Более того, если ты нажмешь на стрелочку около копии, там его можно прямо скачать. Да, да. Ну

соответственно, прямо html файл. А по кнопке publish, ты его можешь опубликовать прямо на сервере entroping.

01:18:54 Но, естественно, без эпина его никто не увидит. Да, да. Но, самая главная суть, что дальше вы можете сказать, что изменить, что кнопку хотите изменить, или цену хотите изменить, или что-то еще. Но, хотелось вам показать, что это, в принципе, непростая задача, если вы не разработчик. То есть, на конструкторе такое не сверстаешь.

01:19:17 Но вот теперь можно такие вещи делать буквально за несколько минут. Более сложные вещи делать, конечно, чуть сложнее. И, опять же, давайте попробуем, сейчас я перейду в Cloud Code, покажу вам просто, как выглядит вот эта машина. Вот он Cloud Code. Соответственно, здесь вы уже работаете с кодом.

01:19:41 Тут надо немножко попрыгать, чтобы всю эту штуку запустить. Опять же, если вы не разработчик, наверное, если вы разработчик, у вас так все уже установлено. Мне пришлось устанавливать там дополнительно WSL на мой компьютер. Это Linux среда внутри вашего Windows компьютера. Опять же, для тех, кто подкован, наверное, это все абсолютно легко, но вот если вы первый раз это делаете, опять же, вы можете спрашивать.

01:20:07 Вот в соседнем окошечке можно спрашивать: Проведи меня пошагово за руку, как мне все это установить? После того, как все это готово, он прямо по технической документации идет клау кода и показывает, как вам все установить. Дальше он подключается к папке вашего проекта. И вот здесь, например, я его попросил: Проанализируй мой проект, найди точки улучшения. Он пошел читать последовательно.

01:20:34 Вот он здесь прочитал сколько-то строк, здесь сколько-то строк. То есть, он зашел в мои папки, в папку проекта, и проанализировал подробно всю структуру. Вот шел по задачам, здесь я вообще ему ничего не говорил, он сам шел, шел, шел и вот говорит. Все, я проанализировал Основные точки улучшения: тестирование критично, покрытие тестами, значит, меньше 10% отсутствует. И вот показывает, куда нужно обратить внимание, что нужно сделать.

01:21:01 Асинхронность и производительность и так далее. Вот он мне написал целую пачку улучшений. Теперь, опять же, будучи не разработчиком, я могу ему написать: А теперь, последовательно, выполни все предложенные тобой улучшения. Понятно, я не разработчик, поэтому я могу позволить себе такую топорную историю, потому что я бы все равно ничего более

содержательного, с точки зрения технической части, ему не сказал бы. Он говорит: Окей, все понятно.

01:21:33 Я последовательно начну выполнять все предложенные улучшения. Это большая задача, поэтому я создам план и буду выполнять его пошагово. Пошел кодить. Он сейчас думает, и вот он первое, что делает, настраивает структуру тестов. Дальше он меня спрашивает, можно ли ему выполнить эти действия, я говорю ему: да, можно, и всё, он пошел это делать.

01:21:55 Как это можно использовать в своей простой прикладной практике, в жизни? Есть у вас какая-то идея, какой-то эксперимент, возможно, вы давно вынашиваете, но надо его прямо кодить что-то, надо каких-то разработчиков искать. Вы можете какого-нибудь простенького бота сделать. Довольно легко, telegram-боты или еще что-то. Вот, как раз, те вопросы, которые по интеграции возникали у вас, их можно вот таким образом решить.

01:22:28 То есть, у вас есть перед руками инструмент, который позволит вам довольно быстро воплотить ваши идеи до MVP, и если вы не знаете, как это сделать, вы можете в нейронку, собственно, тоже сходить и нейросеть, но, по крайней мере, нам очень нравится Клод с точки зрения кодинга. На мой взгляд, наверное, это 1 из лучших сейчас инструментов. Из России немножко проблематично к нему доступ получить, поэтому, если вдруг у вас так и не получится, можно сделать курсоре. Курсор тоже будет хорош для каких-то простых задач. Наверное, можно и таким путем пойти.

01:23:03 Дальше вы просто пишете ему вашу задумку, вашу идею, а он технически описывает ее, реализовывает, и вы ее пробуете как-то запустить и протестировать свою идею, свою гипотезу. Но вот, наверное, первое, с чего можно начать, а то, что касается интеграций, например, вам нужно присоединить его к телеграму. У телеграмма есть API это такая, скажем, розетка, которая торчит от телеграмма наружу и к нему можно подключиться, свою вилку вставить в него. Для того, чтобы это сделать, вы можете дать нейронке техническую документацию, прям выгрузить техническую документацию, сказать: я ничего не понимаю, вот тебе техническая документация, помоги мне подключиться к API вот этой программы, да, с интегрировансья между ними. И нейронка проанализирует техническую документацию, напишет какой-то код для интеграции, который вы сможете у себя запустить.

01:23:57 То есть, в общем, вы можете, если простым языком, вы можете использовать нейросеть не только как условного разработчика, который будет писать код, но и как наставника, который вам поможет какие-то первые действия сделать, потому что нейросеть не может сама, условно, что-то про

инсталлировать на ваш компьютер, там вот эти все вещи сделать это вам нужно сделать самостоятельно. Но сделать вы это можете с помощью нейросетей, запрашивая у нее совета пошагово и она вам поможет, проведет вас за руку в этом процессе. Но вот я показал буквально в деталях, как это работает с точки зрения написания лендинга. Я не знал заранее, какого результата это был экспромт, но на мой взгляд получилось, вот еще раз покажу, вообще очень круто такой лэнд, как бы действительно, ну надо его как-то там дизайннить, все с градиентами, в общем, молодец, кот. Вот давайте поотвечаем немножко на вопросы, Скажите, пожалуйста, есть ли они у вас?

01:25:00

Может быть вопрос ничего не понятно, с чего начать или, не знаю, или наоборот какие-то более глубокие технические вопросы, тогда тут вот как раз Сергей сможет подраскрыть какие-то детали. В общем, дайте нам немножко фидбэка. Здесь огоньков нет, поэтому можете просто написать в чат какие-то ваши мысли, вопросы по теме. Вот, Кристина пишет. Кристина ждет повышения теперь.

01:25:32

Да, повторюсь, где можно найти будет ссылки полезные. Да, давайте мы прямо сейчас это сделаем тоже, ссылок поскидываем, это не проблема. То, что касается depresearch, давайте быстренько вам покажу тогда, прямо вот, кстати, вот, собственно, ваш вопрос: где найти ссылки полезные? Давайте, мы на нем и оттестим perplexity. И я ему пишу: мне необходимо найти инструменты разработки для вайп-кодинга.

01:26:16

Я слышал о cursor и Cloud код, сделай подробный Выключи глубокий удобный ресеоч для этого вопроса, он не обязателен, то есть может он прекрасно с ним справиться и без него, на первом переключателе. Ну да, да, я понимаю, я хотел скорее просто показать, что на самом деле, может быть, думаешь он надолго у тебя уйти минут на 15-20, в принципе. Хотя, может, в принципе, и сам вопрос. Запрос вроде простой, но если вдруг уйдет, я тогда покажу в Клоде, как он ищет что-то, да. Давайте попробуем, просто, чтобы была возможность показать deep research.

01:27:10

Он начал размышлять. Для более быстрого ответа пропустить сбор источников. Для глубокого ответа я проведу поиск по инструментам. Вот он пошел, значит, посмотрел. Курсор, best practice.

01:27:25

Вот видите, он пошел по разным ссылкам, говорит: Я ищу разные-разные варианты. Вот еще 8 источников проанализировал. Двигается дальше то есть он двигается пошагово и подробно разбирает читает все это самаризует с разными поисковыми запросами вот видите поиск-то с таким запросом, поиск с таким, с таким. И каждый вариант дает какой-то результат ему.

Далее он все это вместе собирает и выдаст нам, условно, вот какой-то результат.

01:27:59 Сейчас посмотрим, сколько времени займет. Заранее, действительно, проблем. Мы не знаем точно всегда заранее, сколько времени займет тот или иной анализ. Но качество этого анализа действительно на порядок выше, чем если самому идти искать. То есть, вот он прям конкретные вещи разбирает.

01:28:22 Пока он Вот, собственно, закончил относительно быстро. И вот он говорит, значит, лидеры рынка, вот он, говорит, курсор, действительно, Cloudcode дает следующий обзор. Вот еще Codeon winsurve edition github copilot и альтернативы и видите он везде дает ссылки Вот они ссылки на источники. Вот это и есть depresearch, когда вы приходите с каким-то вопросом и получаете вот такой ответ. Тот же самый perplexity доступен условно бесплатно.

01:29:09 То есть вы можете какое-то количество запросов сделать в день 3 depresearch бесплатно или какое-то количество обычных запросов. Вот, ну и соответственно у нас их довольно-таки много, и по качеству, я так скажу, зачастую им пользоваться гораздо удобнее, чем Гуглом. Да, вот он, видите, сравнение по ценообразованию сделал сравнительную таблицу годовая стоимость 100 разработчиков. Вот он подробно описывает, какой вариант. И есть еще связанные запросы, можно было бы пойти и еще что-то сделать похожее.

01:29:46 Примерно так работает depresearch. Прямо сейчас что я сделаю? Я возьму вот эту штучку, так экспортировать в pdf давайте вот так поделиться каждый у кого есть ссылка оба вот такой вот экспромт Вот, посмотреть, у вас откроется он, интересно, или нет. Потому что Perplexity вроде бы в России не подпадал под какие-то жесткие блокировки. Вот интересно, откроется ли он.

01:30:14 Плюсните в чатик, если у вас открылась эта ссылка. Все есть. Супер. Вот. Ну, что, давайте какие-то может быть есть у вас вопросы, мы их поразбираем еще, как бы несмотря на то, что ровно полтора часа завершились, но вот может быть что-то вам бы хотелось.

01:30:35 Постарались сделать сегодняшний вебинар с разных сторон разобрать эту тему и со стороны вот из разряда. Мне, как человека такого не очень подкованного с точки зрения технической разработки, но при этом я как продакт запускал разные продукты, эксперименты. Например, для меня с точки зрения тестирования гипотез MVP это просто подарок. Потому, что вы можете очень быстро какое-то ваше MVP решение реализовать и получить

на него фидбэк. Я видел у Андрея был вопрос: Давайте мы заодно поэкспериментируем, раз уж мы сегодня и нашу платформу тестируем вебинарную новую.

01:31:21 Давайте дадим слово Андрею и попробуем сейчас запросить доступ к микрофону. Давайте мы так повысит, как ведущего. Давайте посмотрим, Андрей, так повысит, как ведущего и разблокировать микрофон. Вот, Андрей, ну-ка скажите что-нибудь. Мы тестируем с вами в real-time функционал.

01:31:50 Так попробуйте нажать включить микрофон. Возможно, у вас получится с нами пообщаться, раз уж вы руку подняли. Так разблокировать веб-камеру. Так, у Андрея пока Андрей, тогда если не получается вдруг включится или если неудобно, то можете написать в чатик. Я попробовал вас включить, вывести в эфир.

01:32:23 Но вот что-то пока либо у нас, либо у вас, что да. Нужно включить микрофон, выбрать и нажать еще раз на микрофон. Тогда у вас получится. А у Сергея пока можно полюбоваться на прекрасного кота, который тоже вышел к нам в эфир без спроса и без поднятия рук. Она помогать пришла.

01:32:48 Да, Вайп-кодить. Вайп-кодить. Андрей снова поднял руку. Андрей, мы сделали все, что могли, чтобы вывести вас в эфир. Если у вас есть возможность, то включайтесь.

01:33:04 Нужно выбрать камеру, микрофон и включить. Если у вас не получается, напишите в чатик, мы с вами пообщаемся тогда таким образом. Раз, 2, 3 тест микрофона. Вот, отлично. Андрей, это вы?

01:33:25 Это не Андрей, это Гудков Александр, где я просто решил проверить все ли работает. Да, да, мы вот только что разблокировали возможность. Да, отлично. Ну что, тогда, если есть вопросы, пожалуйста, задайте их чат или поднимите руку, если хотите задать голосом. Если вопросов нет, то тогда мы просто завершим.

01:33:47 Надеюсь, что мы сегодняшнюю тему раскрыли. Дайте нам, пожалуйста, фидбэк еще, помимо прочего, как вам сегодняшний вебинар, как вообще сегодняшняя тема, что хотелось бы раскрыть возможно более детально или более простыми словами где-то, может быть мы технически слишком сложно сегодняшнюю тему развернули и нужно начинать вообще с понятия, что такое сервер, что такое код, что это не код, который у Сергея, который вот сейчас на руках у него мягкий пушистый, а что это код с буквой д на конце. Вот, В общем, скажите нам, пожалуйста, какой-то фидбэк нам будет очень,

потому что эта тема сегодня тоже была экспериментальная, такая она более техническая. У нас есть идея, где такая гипотеза, что кому-то может быть интересно даже курс по этой теме пройти и, возможно, мы его, собственно, сделаем тогда. В общем, интересно, что хотелось бы еще, может быть, раскрыть как-то детально.

01:34:49 Вот вижу, пишут, что очень понравилось, все было супер, спасибо вам большое, мы старались. И еще интересно, пользуясь случаем, как вам в вебинарной системе, которая сегодня была, в отличие от МТС линка, сегодня мы использовали свою вебинарную платформу и вот интересно, как ваш опыт взаимодействия с ней, удобно ли было вообще слушать вебинар сегодняшний, смотреть его. Была ли удобно видна демонстрация экрана, чат. В общем, дайте нам и в этой части тоже обратную связь, пожалуйста. Мы хотим перейти полностью на свою вебинарную систему, поэтому интересно, как вам сегодня.

01:35:32 Так вот Никита тоже написал очень круто, спасибо большое. Так спасибо и очень интересно с учетом моего образования, включая навыки программирования. Вот супер, спасибо большое, Александр. Только концу семинара разобрался. Ага.

01:35:57 Да, видно, слышно было хорошо. Есть мелкие неудобства, чат наезжает на input ввода, картинка стабильная, вылетов не было, звук не прервался. Ура! Мы тоже так себя чувствуем. Ура!

01:36:13 Так, хорошо. Супер! Спасибо большое! Сама платформа намного удобнее, чем МТС. Да, интересно, так бывает.

01:36:23 Надеемся, что мы еще ее доработаем и будет вообще удобно. Тут, на самом деле, очень много возможностей. МТС не помню, но тут удобно вроде кроме, что чат внизу наезжает, темную тему надо. Ага, Хорошо. Это мы тоже сделаем.

01:36:38 Так, обещанный подарок. Говорил я про него сегодня и не показал вам его. Надо бы выполнить свои обещания. Сейчас, 1 секундочку, я открою презентацию, то, что меня выкидывало. Для тех, у кого вдруг нет доступа к Alpina GPT, и, поскольку вы с нами до конца сегодняшнего вебинара пробыли, сейчас я вам покажу на экране этот промокод.

01:37:04 И Rezidda его как раз тоже скинула в чате. Спасибо большое! Так, а я открываю презентацию, и на экране, внимание, появляется qr-код. Не тот, который нужно, а тот, который нужно сейчас появится. Да, если что, Reseda уже скинула, Вот, ну в общем, наверное, я опоздал с qr-кодом.

- 01:37:32** Пусть пускай будет. Так и почти дошли. Вот он. Можете навести камеру, отсканировать, перейти. Кстати говоря, ридида скинула не ту ссылочку на регистрацию.
- 01:37:48** Давайте-ка я сейчас быстренько скину актуальную новую последнюю. Так потому что по QR-коду у вас как раз перейдёт переведёт туда, куда нужно. Так, сейчас, 1 секунду. Так. Потому что мы подготовили для вас новую версию платформы.
- 01:38:15** Вот она. Вот по ней, пожалуйста, перейдите и зарегистрируйтесь. И вот здесь вы сможете, да. Сможете, собственно, ввести свои данные для регистрации. Да.
- 01:38:37** Так. Ну что, тогда мы сегодня будем завершать, учитывая, что вопросов вроде нет, только руку поднимают. Если поднимаете руку, наверное, вы что-то хотите сказать, скажите или напишите в чат, или давайте мы вам дадим возможность случайно. Если что, тут можно поднимать руку, а мы на это реагируем. Так, ну что, спасибо вам большое!
- 01:39:03** Серег, может быть ты хотел что-то еще в конце, какое-то напутственное слово? Настя написала, что интересно, и рада была узнать, что ее не заменит. Разработчикам пока переживать не о чем. Может быть, хотел бы дополнить в конце, и мы завершим сегодняшнюю работу. Вопросы использовать или не использовать, ну, как по мне, решенный, то есть, использовать, да, аккуратно, да, под обзором, но использовать, то есть, я как человек, как ты сказал, с более 20 лет опыта, соответственно, как бы, с разработкой застал те времена, когда вообще еще не было ни IDE, как бы, ничего.
- 01:39:43** Писали код в блокнотике, но это же не означает, что там должны были остаться, потом пришли какие-то редакторы, IDE и прочее-прочее, то есть мир развивается, как бы, и нужно быть с ним на 1 волне и развиваться, поэтому ИИ это уже, мне кажется, шаг в будущее, который определенный и от этого уже никуда не уйти, то есть просто нужно адаптироваться и учиться его использовать. Да, как и во всех остальных сферах жизни. Я начал использовать нас. Да, и как во всех остальных сферах жизни примерно похожая история. Спасибо большое.
- 01:40:22** Тогда мы на сегодня будем завершать. Желаем всем хорошего вечера, хороших выходных и увидимся с вами в ближайшее время на новых вебинарах по теме искусственного интеллекта, которые мы сейчас в Альпине активно раскачиваем. Всем пока! Спасибо большое! Спасибо, Серег, тебе за то, что вышел и так рассказал очень подробно про эту тему.

01:40:43

Спасибо. Пока-пока.

Жемал Хамидун · hamidun.com · @jemaL_hamidun