

БОТЫ

без лишнего КОДА

Как создать чат-бота
для бизнеса и личных задач



Марк Тьюрин

Марк Тьюрин

**Боты без лишнего кода:
Как создать чат-бота для
бизнеса и личных задач**

«Автор»

2026

Тьюрин М.

Боты без лишнего кода: Как создать чат-бота для бизнеса и личных задач / М. Тьюрин — «Автор», 2026

Сколько раз за день вы отвечаете на одинаковые вопросы, уточняете записи и отвлекаетесь от работы ради сообщений? Бот может снять эту рутину — если решает конкретную задачу, а не становится ещё одним каналом, за которым нужно следить. «Боты без лишнего кода: Как создать чат-бота для бизнеса и личных задач» проведёт вас от идеи до запуска: поможет найти узкое место, выбрать задачу для первой версии, спроектировать сценарий и собрать бота в конструкторе без программирования. Вы разберётесь, как выбрать площадку, связать заявки с рабочим процессом, писать понятные сообщения, собирать только нужные данные и вовремя передавать диалог сотруднику. Тестирование, пилот, оценка результата, поддержка и безопасное отключение — части одного процесса. Книга адресована владельцам малого бизнеса, самозанятым и специалистам без опыта в разработке, а также тем, кто хочет автоматизировать личные дела.

© Тьюрин М., 2026

© Автор, 2026

Содержание

Бот решает задачу, а не украшает бизнес	5
Где теряется время и заявка	11
Кому, где и в какой момент нужен бот	18
Маленький запуск вместо вечного проекта	24
Сценарий как карта разговора	29
Конец ознакомительного фрагмента.	34

Марк Тьюрин

Боты без лишнего кода: Как создать чат-бота для бизнеса и личных задач

Бот решает задачу, а не украшает бизнес

К полудню в мастерской «Спица» уже пахнет смазкой и мокрым асфальтом. Ирина держит колесо одной рукой, а другой снимает перчатку, чтобы ответить на сообщение: «Сколько будет стоить устранить люфт? И можно привезти велосипед завтра?» Точную стоимость она назовёт после осмотра, а свободное время нужно сверить с Павлом. Ирина пишет об этом, кладёт телефон экраном вниз и возвращается к работе.

Через несколько минут звонит телефон, затем приходит сообщение о готовом велосипеде и вопрос о наличии камер. Павел в это время принимает велосипед у входа и записывает данные клиента. Позже Ирина открывает переписку и видит ответ Марины: «Завтра в десять подойдёт». Но успел ли кто-нибудь подтвердить время? В календаре имени Марины нет. Сообщение не пропало — оно затерялось среди других, а договорённость так и осталась между предложением и записью.

Ирина не хочет пропускать обращения и отвечать с телефоном в руках, пока держит инструмент. Павел опасается другого: если добавить бота, придётся следить ещё и за ним. Оба говорят о реальной проблеме, но пока называют её по-разному. Ирина хочет освободить руки, Павел — не получить новый источник путаницы.

Сначала задача, потом инструмент

Разговор о боте часто начинается с вопроса: «Где его сделать?» Но конструктор не подскажет, что именно поручить автоматике. Если сначала не разобраться в процессе, бот быстро и одинаково ответит на ненужные клиенту вопросы, соберёт данные, которые никто не обрабатывает, или подтвердит запись, которой на самом деле нет.

Чат-бот без кода — это настроенный сценарий общения в чате. Он задаёт вопросы, выдаёт подготовленные ответы, передаёт информацию или запускает действия по заданным правилам. Бота собирают из готовых блоков в визуальном конструкторе, а не программируют с нуля. Простой сценарий может выглядеть так: человек выбирает услугу, указывает удобный день, оставляет контакт, получает подтверждение или сообщение о следующем шаге.

Такой бот не обязательно использует искусственный интеллект. Меню с кнопками и понятными условиями — уже бот. И наоборот: если сотрудник вручную отправляет заготовленные сообщения, это ещё не полноценная автоматизация, хотя со стороны может казаться, что ответы приходят «системно». Для решения практической задачи важнее не название технологии, а то, кто или что выполняет следующий шаг.

В повторяющемся процессе обычно встречаются три вида работы.

Повторяющийся вопрос — запрос, на который можно дать один и тот же точный ответ, пока не изменились условия. Например: где находится мастерская, когда она открыта, сколько стоит диагностика, какие данные нужны для записи. Важно не только то, что вопрос задают часто: ответ должен оставаться стабильным. Если цена зависит от состояния велосипеда, бот не должен называть стоимость ремонта, зная лишь название неисправности.

Операция — действие по понятным правилам: принять заявку, записать контакт, распределить обращения по типам, отправить напоминание. В отличие от ответа, операция меняет ход процесса: из сообщения появляется заявка с датой и описанием проблемы.

Решение — выбор следующего шага в зависимости от ответа. Если клиенту нужна настройка переключателя, бот может предложить подходящее время. Если речь о повреждённой раме или неисправных тормозах, лучше передать обращение сотруднику. Бот может следовать заданным условиям, но не должен изображать уверенность там, где мастер ещё не осмотрел велосипед.

Эти три вида работы часто соединяются: бот отвечает на типовой вопрос, собирает сведения о заказе и направляет человека дальше. Но не каждому процессу нужны все три части. Иногда достаточно одного точного ответа, иногда — ссылки на форму. Бот полезен там, где последовательность действий повторяется, а автоматизация действительно снимает лишнюю работу.

Путь Марины — от первого вопроса до записи

Павел предлагает не проектировать бота целиком, а проследить путь одного клиента. Они берут сообщение Марины и разбирают, что должно произойти, чтобы она получила ясный ответ, а мастерская — корректную запись.

Первый шаг — обращение. Марина спрашивает о стоимости и времени. Автоматизировать здесь можно только то, что мастерская знает заранее. Бот способен сообщить часы работы и адрес, назвать цену диагностики, если она установлена, и объяснить, что окончательная стоимость ремонта зависит от осмотра. Если стоимость меняется в зависимости от обстоятельств, не нужно прятать ответ за меню. Лучше сразу обозначить границу: «Диагностика стоит по действующему прайсу. Стоимость ремонта назовём после осмотра».

Такой ответ экономит время и не создаёт ложных обещаний. Клиенту проще решить, стоит ли продолжать, а мастерской не приходится разбирать по переписке каждую возможную неисправность.

Второй шаг — уточнение услуги. Фраза «Нужен ремонт» слишком широка: человеку может требоваться замена камеры, регулировка тормозов или проверка колеса. Павел предлагает короткий список основных категорий и поле для свободного описания. Ирина добавляет важное ограничение: список нужен для маршрутизации, а не для удалённой диагностики. Если клиент отмечает, что тормоза не работают, система не должна сама предлагать обычную запись и тем более уверять, что ехать на велосипеде безопасно. Она может попросить пока не пользоваться велосипедом и передать сообщение мастеру.

Автоматизация помогает собрать исходные сведения, но не заменяет профессиональную оценку. Клиенту не нужно угадывать технический термин. Достаточно выбрать «тормоза», «колесо», «переключение передач» или «другое», а затем описать симптом своими словами. Если описание останется непонятным, сотрудник задаст уточняющий вопрос.

Третий шаг — выбор времени. Здесь маршрут особенно уязвим. Ответ «завтра в десять подойдёт» означает, что клиенту удобно предложенное время, но ещё не обязательно, что мастерская внесла запись в расписание. В случае Марины именно это различие осталось неясным.

Если бот показывает свободные интервалы из актуального расписания, он может предложить время и сразу внести его в календарь. Тогда запись действительно подтверждена автоматически. Но если расписание ведут в тетради или обновляют вручную, бот не знает, что только что произошло в мастерской. Он может принять пожелание, но не должен сообщать «Вы записаны», пока сотрудник не проверил, есть ли свободное окно.

Для первого запуска Павел предлагает честную формулировку: «Вы отправили заявку на завтра, 10:00. Администратор проверит расписание и подтвердит запись сообщением». Ирина уточняет: клиенту нужно назвать срок ответа — например, в рабочее время в течение часа, если команда действительно может его соблюдать. Обещание «скоро ответим» не помогает планировать ни клиенту, ни сотруднику.

Четвёртый шаг — подтверждение. После проверки расписания клиенту нужно отправить короткое сообщение с датой, временем, видом услуги и адресом. Если запись пока предварительная, так и следует написать. Можно добавить возможность перенести или отменить визит, чтобы изменение не затерялось в переписке. Напоминание перед визитом полезно, только если в нём указаны верные дата и время и клиент понимает, как сообщить об изменении планов.

В итоге карта процесса «Спицы» выглядит так.

Обращение: на типовой вопрос о цене, часах работы и порядке записи можно ответить автоматически. Нестандартное описание неисправности может потребовать участия сотрудника.

Уточнение услуги: выбрать категорию и собрать описание проблемы можно по повторяющемуся сценарию. Причину неисправности и объём ремонта определяет мастер.

Выбор времени: свободный интервал можно показать и внести в календарь, если расписание актуально. Если время проверяют вручную, бот принимает только пожелание клиента.

Подтверждение: отправить детали записи и напоминание можно автоматически. Изменение срока ремонта или срочную проблему нужно передать человеку.

У этой карты есть важное свойство: каждый шаг заканчивается понятным результатом. После первого клиент знает условия. После второго мастерская понимает, о чём обращение. После третьего либо есть запись, либо клиент знает, что время ещё проверяют. После четвёртого обе стороны видят одну и ту же договорённость.

Что автоматизация снимает, а что добавляет

Когда Ирине приходится по несколько раз за день отвечать на один и тот же вопрос, дело не только в потраченных на набор минутах. Нужно отложить инструмент, найти сообщение, восстановить контекст, проверить условия, написать ответ и снова вспомнить, на чём остановилась работа. Короткое действие разрывает длинное.

Если типовых вопросов много, бот или даже готовый ответ сокращает число таких переключений. Если в каждой заявке нужно собрать одни и те же сведения, автоматизированный сбор избавит от вопросов вроде «Какой у вас велосипед?», «В чём проблема?» и «Когда удобно приехать?». А автоматическое напоминание снижает риск, что заявка останется без ответа.

Но автоматизация не всегда уменьшает нагрузку. Боту нужны актуальные ответы, проверка расписания, обработка исключений и понятная передача диалога сотруднику. Если Павлу придётся следить за двумя каналами вместо одного, вручную сверять каждую запись и исправлять ошибки в подтверждениях, работы станет больше. Поэтому до запуска важно оценить не только число сообщений, но и то, какой именно ручной шаг исчезнет.

Для простой проверки можно в течение недели отмечать, сколько раз команда отвечает на один и тот же вопрос и сколько времени занимает обработка заявки. Допустим, вопрос о цене диагностики задают двенадцать раз в день, а ответ занимает минуту. Это уже двенадцать минут набора. Но если после ответа сотрудник ещё несколько минут уточняет вид ремонта и свободное время, значительная часть нагрузки связана не с самим ответом, а с дальнейшими шагами. Закреплённое сообщение снимет только часть проблемы.

Бот особенно полезен, когда убирает повторяющиеся действия и не требует новых ручных проверок. Он мешает, если заставляет клиента проходить длинную цепочку ради простого ответа, пытается загнать нестандартный разговор в меню или выдаёт предварительную заявку за подтверждённую запись. А когда становится ясно, что нужен человек, бот не должен продолжать задавать вопросы.

У Ирины есть причина не автоматизировать всё подряд. Если Марина пишет: «После падения колесо стало криво, можно доехать до вас?», быстрый ответ с кнопками не обязательно будет лучшим решением. Здесь важнее осторожность, а не скорость. Мастер может уточнить обстоятельства или посоветовать доставить велосипед другим способом. Бот способен обозна-

чить границу своих возможностей и переключить диалог на сотрудника, но не должен вести себя так, будто осмотр уже состоялся.

В других случаях человеческий разговор нужен не из-за сложности технологии, а из-за неопределённости. Клиент может не знать названия детали, писать с ошибками, прислать бесполезную фотографию или сбивчиво описать проблему. Хороший сценарий не наказывает человека за это, а предлагает понятный выход: «Не нашли подходящий вариант? Опишите проблему сообщением — администратор поможет».

Реплики, которые не создают ложных ожиданий

Павел выписывает короткие сообщения, которые пригодятся независимо от того, выберет мастерская бота или форму. Ирина оценивает их не по красоте, а по простому вопросу: что клиент поймёт после прочтения?

Для типового вопроса:

«Диагностика велосипеда проводится в мастерской. После осмотра мастер сообщит, какие работы нужны и сколько они будут стоить. Если хотите записаться, выберите удобный день или отправьте заявку».

Если у диагностики фиксированная цена, её стоит назвать прямо и проверить, что она соответствует действующему прайсу. Если стоимость зависит от вида услуги, лучше объяснить, что можно рассчитать заранее, а что — только после осмотра.

Для заявки без автоматического доступа к расписанию:

«Укажите желаемые день и время. Мы проверим расписание и напомним, удалось ли закрепить этот интервал. До подтверждения заявка не считается записью».

Для автоматической записи при синхронизированном расписании:

«Вы записаны на 14 июня, 10:00, на диагностику велосипеда. Если планы изменятся, ответьте на это сообщение, чтобы перенести или отменить визит».

Для нестандартного случая:

«По описанию нельзя надёжно определить причину неисправности. Передам сообщение мастеру. Если проблема связана с тормозами или повреждением рамы, не используйте велосипед до осмотра».

Фразы должны соответствовать реальному процессу. Если мастерская не проверяет сообщения вечером, нельзя обещать ответ в любое время. Если запись создают вручную, нельзя сообщать, что клиент уже записан. А если заявка уходит в общий чат, но никто не отвечает за её обработку, слово «передам» лишь маскирует проблему.

Павел предлагает добавить кнопку «Позвать сотрудника». Ирина спрашивает, кто будет отвечать на такие обращения. Это не мелочь интерфейса, а часть сценария. У кнопки должен быть ответственный — сотрудник, который увидит сообщение и знает, когда на него нужно ответить. Если такого человека нет, лучше честно указать порядок: «Оставьте сообщение, администратор ответит в рабочее время». Бот не должен создавать впечатление, что кто-то уже взялся за обращение.

Выбрать между ботом, формой и готовым ответом

После разбора процесса Ирина и Павел сравнивают три простых решения. Они выбирают не самое технологичное, а то, которое устраняет обнаруженный сбой.

Готовый ответ подходит, если вопрос повторяется, ответ не меняется и собирать сведения или запускать дальнейшие действия не нужно. Например, так можно сообщить адрес мастерской, часы работы, цену диагностики или объяснить, как подготовить велосипед к приёму. Такой ответ не запишет человека, зато избавит от десятков одинаковых объяснений.

Форма подходит, если нужно собрать несколько сведений, а последовательность вопросов почти не меняется. Человек указывает имя, контакт, вид услуги, описывает проблему и предлагает удобное время. Сотрудник получает аккуратную заявку и связывается с ним. Форме

не обязательно имитировать разговор: многим клиентам быстрее заполнить пять понятных полей, чем проходить цепочку сообщений, нажимая «Далее».

Бот подходит, если клиенту нужно пройти по разным веткам, получить подходящие ответы, выполнить действие или узнать, что происходит с заявкой. Например, выбрать вид ремонта, получить следующий шаг, оставить заявку, проверить её статус или получить напоминание. Чем больше условий и действий, тем полезнее сценарий — но тем важнее убедиться, что он не обещает того, чего мастерская не может выполнить.

Решение можно принять, ответив на три вопроса.

1. Нужно только дать повторяющийся ответ? Начните с готового ответа. Если сотрудники каждый раз набирают его заново или клиенту трудно его найти, разместите сообщение там, где чаще всего задают этот вопрос.

2. Нужно собрать один и тот же набор сведений? Начните с формы. Она особенно уместна, когда человек может заполнить всё за один раз, а дальше с заявкой работает сотрудник.

3. Нужно направить клиента по разным веткам, выполнить действие или сообщать ему о состоянии заявки? Рассмотрите бота. Проверьте, где хранится расписание, кто разбирает исключения и что получит клиент, если сценарий ему не подойдёт.

Есть и ещё одно правило: если ошибочное автоматическое решение может стоить клиенту денег, сорвать визит или повлиять на безопасность, оставьте это решение человеку. Автоматизировать можно сбор информации и передачу обращения — но не обязательно заключительный вывод.

Например, репетитору, которому каждый день задают вопрос о стоимости занятия, может хватить закреплённого ответа. Если клиенты выбирают предмет, уровень, формат и время, а затем получают разные варианты записи, пригодится бот. А семье, которая один раз собирает ответы на вопрос «Кто что привезёт на дачу?», чаще хватит общей формы или таблицы. Отдельный бот для разовой координации может потребовать больше внимания, чем сам процесс.

Короткая проверка перед запуском

Чтобы не начинать с конструктора, сформулируйте проблему одним предложением: «Когда происходит [событие], человеку или команде приходится [лишнее действие или сбой], из-за чего [конкретное последствие]». Для «Спицы» получится так: «Когда клиент пишет о ремонте, Ирина и Павел вручную уточняют одни и те же сведения и сверяют время в разных местах, поэтому заявки могут оставаться без подтверждения».

Затем опишите желаемый результат: «После первого обращения клиент должен понимать условия и либо получить подтверждённую запись, либо знать, что мастерская ещё проверяет время». Так станет ясно, что считается завершённой работой. «Бот отвечает быстрее» — расплывчатый результат. «Клиент отличает заявку от подтверждённой записи» — проверяемый.

Теперь отделите повторяющиеся ситуации от исключений. Типовые ответы, сбор нужных сведений и напоминания можно описать правилами. Для исключений — неясной неисправности, срочного вопроса или отсутствия подходящего времени — нужен понятный маршрут к человеку. Если вы не знаете, кто подхватит такое обращение, сценарий пока не готов.

Перед настройкой задайте себе три вопроса: уменьшится ли конкретное ручное действие, получит ли клиент ясный следующий шаг и сможет ли команда поддерживать ответы и разбирать случаи, с которыми бот не справился? Если хотя бы на один ответ «нет», сначала поправьте процесс или выберите более простое решение.

Ирина решает не запускать автоматическую запись сразу. Сначала «Спица» подготовит точные ответы на частые вопросы и форму заявки. Павел проверит, какие сведения нужны мастеру, а Ирина договорится с ним, как подтверждать свободное время. Если выяснится, что клиенты регулярно проходят один и тот же путь, а обработка заявок всё ещё требует несколь-

ких ручных действий, мастерская добавит бота — с передачей нестандартных случаев сотруднику.

Такой порядок помогает не строить удобную оболочку вокруг неясного процесса. В следующей главе разберём, как находить места, где теряются время и заявки: не по общему ощущению загруженности, а по конкретным паузам, повторным вопросам и незавершённым переходам между шагами.

Где теряется время и заявка

Бот выбрали не ради красоты и не как ещё один канал связи. Его рассматривают как способ снять повторяющуюся работу и довести человека до ясного результата. Но даже чётко сформулированная задача пока лишь гипотеза: прежде чем что-то автоматизировать, нужно понять, где процесс действительно буксует. Ирина, владелица мастерской «Спица», была уверена: ей не хватает новых заявок. Переписка показала, что дело, возможно, в другом.

Утром в мастерской звонил телефон, у верстака ждал велосипед с погнутым колесом, а Павел, администратор, отвечал на сообщения между разговорами с клиентами у стойки. Ирина посмотрела на пустые окна в расписании и сказала:

— Нужно привлекать больше людей. Обращений должно стать больше.

Павел возразил осторожно:

— Я помню несколько случаев, когда люди спрашивали про ремонт, а потом мы не могли найти переписку.

Это звучало правдоподобно, но ещё не было доказательством. Ирина помнила клиентку, которая, кажется, ушла к конкурентам. Павел — сообщения, оставшиеся без ответа. Оба могли быть правы, но отдельные воспоминания не показывали, как часто случается сбой и насколько он мешает работе мастерской.

Аналитик предложил не спорить о причинах, а восстановить путь последних обращений: от первого сообщения до записи, отказа или неизвестного исхода. Для начала решили проверить не одну историю, которую все хорошо помнили, а последние двадцать четыре обращения из разных каналов.

Версии, которые звучат убедительно

«Нам просто нужно больше заявок». Если на рекламу и новые каналы придёт больше людей, а старые обращения по-прежнему будут обрабатываться с задержкой, нагрузка вырастет, а сбой никуда не денется. Сначала стоит выяснить, сколько людей уже обращаются, сколько получают ответ и что происходит дальше.

«Мы отвечаем быстро». Сотруднику легко вспомнить удачные переписки: клиент написал, администратор сразу увидел уведомление и ответил. Но сообщение могло прийти в разгар работы, затеряться среди звонков или попасть в канал, который проверяют реже. Ощущение скорости — не то же самое, что время, которое показывают отметки в переписке.

«Если клиент не ответил, значит, он передумал». Иногда так и бывает. Но человек мог ждать, получить неполный ответ, перейти в другой канал или договориться по телефону. Молчание клиента не доказывает ни потерю заявки, ни успешное завершение общения. Если исход неизвестен, его так и нужно записать.

«Бот автоматически устранил ошибки». Он может собирать сведения в одном формате и не забывать задать обязательный вопрос. Но если мастерская не знает, какие данные нужны для записи, или расписание ведётся в нескольких местах, бот ускорит передачу неверной информации. Автоматизация ускоряет процесс, но сама по себе не исправляет плохие правила.

Все эти версии полезны как гипотезы. Проверка начинается с того, что можно увидеть: временных отметок, сообщений, записей и действий сотрудников.

Следы вместо воспоминаний

Павел и аналитик взяли последние двадцать четыре уникальных обращения за десять рабочих дней. «Уникальным обращением» считали первый запрос одного клиента по конкретной задаче. Если человек сначала написал в сообщество мастерской, а через час позвонил насчёт того же ремонта, это не две новые заявки, а одно обращение с переходом между каналами. Повторный запрос по другому ремонту — уже отдельный случай.

В журнал не вносили имена и телефоны клиентов. Каждому обращению присвоили условный номер, а из переписок сохранили только сведения, нужные для анализа. Так можно обсуждать сбои в процессе, не превращая аудит в коллекцию личных сообщений.

Для каждой строки записывали канал, время первого сообщения, время ответа, результат, ручные действия, ошибку или потерю. Ответом считали не автоматическое «Здравствуйте, мы скоро ответим», а первое сообщение, которое давало нужную информацию или понятный следующий шаг.

Канал	Время первого сообщения	Время ответа	Результат	Ручные действия	Ошибка или потеря
Сообщество мастерской	09:12	09:46	Запись на диагностику	Данные перенесены в блокнот и календарь	Одна и та же запись внесена в два места
Телефон	10:03	10:07	Запись подтверждена	Сведения записаны в журнал, затем в расписание	Не указан тип ремонта, пришлось уточнять
Сообщения	12:26	13:51	Получен ответ о сроках	Павел вручную проверил загрузку мастера	До ответа клиент ждал больше часа
Форма на сайте	16:38	17:52	Клиент сообщил, что записался в другом месте	Уточняли, можно ли принять велосипед на следующий день	Клиент прямо связал решение с ожиданием
Сообщество мастерской	17:42	09:14 следующего дня	Вопрос о статусе ремонта решён	Павел уточнил сведения у механика	Вопрос пришлось искать между новыми сообщениями
Телефон	14:11	15:22	Итог не восстановлен	Был пропущенный звонок, затем попытка перезвонить	Неизвестно, удалось ли связаться с человеком

Это лишь несколько строк из журнала, а не все двадцать четыре. Они показывают, почему недостаточно записать только «заявка пришла» или «заявка не пришла». Важно и то, что происходило между этими отметками.

В одном случае клиенту ответили быстро, но данные дважды внесли вручную. В другом ответ занял больше часа, хотя сам он был коротким: Павел сначала отложил вопрос, а потом уточнил расписание. В третьем исход остался неизвестным. А в случае с формой на сайте есть не просто предположение о потере: клиент написал, что записался в другом месте, потому что ответа пришлось ждать. Это весомее обычного молчания, но всё равно не доказывает, что любая задержка приводит к потере клиента.

За десять рабочих дней мастерская получила двадцать четыре уникальных обращения. Девятнадцать из них начались в рабочее время. Восемь человек спрашивали о нескольких

повторяющихся вещах: стоимости диагностики, сроках и доступности записи. В девяти обращениях сведения пришлось переносить из переписки или телефонной заметки в расписание. В пяти из девятнадцати обращений, начатых в рабочее время, содержательный ответ пришёл больше чем через час. Трижды запись пришлось исправлять или уточнять. В двух случаях восстановить итог общения не удалось.

Категории могут пересекаться. Например, один и тот же клиент мог задать типовой вопрос, долго ждать и в итоге попасть в расписание с неполными данными. Поэтому не стоит складывать категории так, будто речь идёт о разных заявках. Журнал нужен, чтобы увидеть рисунок процесса, а не получить красивую сумму.

Два разных времени

При разборе Павел заметил, что фраза «долго отвечали» скрывает два разных показателя.

Первый — ожидание клиента: сколько прошло от первого сообщения до содержательного ответа. Второй — рабочее время сотрудника: сколько минут Павел потратил на ответ, уточнения, запись и исправления. Между ними может быть большая разница. Павел мог потратить три минуты на короткий ответ, но отправить его через час, потому что сначала закончил разговор у стойки. Или быстро отреагировать на сообщение, а затем ещё десять минут искать расписание и переносить данные.

В журнал можно заносить оба показателя. Ожидание видно по временным отметкам в канале. Рабочее время на первом этапе достаточно оценивать приблизительно: две минуты на ответ, три — на уточнение, ещё две — на внесение записи. Если действие трудно измерить точно, лучше указать диапазон, чем делать вид, будто известна каждая секунда.

Отдельно учитывайте часы работы. Если клиент написал в 17:42, а мастерская закрылась в 18:00 и ответила в 09:14 следующего дня, календарное ожидание составило пятнадцать часов тридцать две минуты. Но в рабочее время попали только тридцать две минуты: восемнадцать до закрытия и четырнадцать после открытия. Клиенту важно календарное ожидание, а для оценки нагрузки на администратора полезно знать и рабочее. Не подменяйте одно другим.

В «Спице» медианное время до содержательного ответа по обращениям, начатым в рабочее время, составило двадцать восемь минут. Медиана — это значение посередине упорядоченного списка: примерно половина обращений получила ответ быстрее, половина — медленнее. Для небольшой выборки она часто понятнее среднего, которое может сильно измениться из-за одного особенно долгого ожидания. Мастерская также отметила, что в пяти случаях из девятнадцати ответ занял больше часа. Эти два числа описывают одну ситуацию, но отвечают на разные вопросы: первое показывает типичное ожидание, второе — как часто возникали долгие паузы.

Цена пропуска и ошибки

Павел предположил, что пропущенный звонок, вероятно, лишил мастерскую заказа. Аналитик предложил разделить три утверждения.

«Обращение не найдено» — наблюдаемый факт, если записи действительно нет.

«Клиент не записался» — тоже факт, если это удалось проверить, например по расписанию и переписке.

«Клиент не записался именно из-за нашей задержки» — уже вывод о причине. Он убедительнее, если человек прямо говорит об этом, но не позволяет автоматически объяснить все случаи отсутствия ответа.

Чтобы оценить цену ошибки, сначала полезно измерить её последствия в том, что можно подтвердить. Сколько раз пришлось перезвонить? Сколько минут ушло на исправление? Заняли ли неподходящий слот? Клиент приехал не в тот день? Механик ждал велосипед, который так и не привезли? Такая проверка показывает цену сбоя, даже если потерянную продажу невозможно доказать.

Если мастерская хочет оценить денежные потери, нельзя умножать число пропущенных сообщений на среднюю стоимость ремонта и называть результат потерянной выручкой. Не каждое сообщение приведёт к заказу, а стоимость ремонта включает расходы на детали и работу. Сначала можно посчитать подтверждённые случаи, когда клиент сообщил, что ушёл из-за ожидания, а затем отдельно оценить сумму, остающуюся после прямых расходов. Неопределённые обращения лучше показывать как возможный диапазон потерь, а не как гарантированный доход, которого мастерская лишилась.

Ручное время тоже не всегда превращается в прямую экономию. Если Павел потратит на однотипные ответы на час меньше, расходы на зарплату в мастерской не обязательно сократятся. Зато у него появится час на подтверждение записей, помощь клиентам у стойки или проверку сложных ремонтов. Это ценность освободившегося времени, а не автоматически сэкономленная сумма.

Ирина хотела узнать, сколько времени можно вернуть. Журнал показал, что повторяющиеся ответы и перенос сведений отнимают время, но пока его фиксировали не в каждом случае. Поэтому на следующей неделе решили отмечать, сколько минут занимает типовой ответ и перенос записи. Если ответ длится от двух до четырёх минут, а перенос — от двух до трёх, можно рассчитать осторожный диапазон. Но складывать показатели без проверки нельзя: одно обращение могло включать и ответ, и перенос, а часть действий автоматизировать не получится.

Узкое место — не то же самое, что удобный сценарий

После сбора фактов каждое повторяющееся действие полезно оценить по трём признакам: как часто оно возникает, к чему приводит и можно ли его автоматизировать.

Частота показывает, сколько раз одно и то же случилось за выбранный период. Лучше сказать «восемь раз из двадцати четырёх», чем «постоянно». Последствия помогают понять, чем действие обходится мастерской: ожиданием, повторным вводом, исправлением записи или подтверждённым отказом клиента. А пригодность для автоматизации зависит от того, можно ли заранее описать правильный ответ и следующий шаг.

Высокая частота сама по себе не делает задачу подходящей для бота. Если люди часто спрашивают, сколько точно продлится ремонт велосипеда, а срок зависит от диагностики и очереди, бот не должен придумывать ответ ради скорости. Он может объяснить, какие сведения нужны, собрать их и передать сотруднику. А на вопрос со стабильным ответом — например, где находится мастерская или как записаться на диагностику, — бот может ответить сразу.

Редкая, но серьёзная ошибка тоже заслуживает внимания, хотя не всегда требует бота. Если неверно записанная дата срывает рабочий день, сначала нужно понять, почему данные расходятся. Возможно, мастерская ведёт два календаря, и ей нужен единый источник расписания, а не сложный сценарий общения. Иногда достаточно решить, кто вносит изменения.

На практике логика проста: частую задачу с ясным ответом стоит проверить на возможность автоматизации. Если задача тоже частая, но требует человеческого решения, бот может собрать исходные данные и передать их сотруднику. Редкий, но дорогой сбой — повод отдельно разобраться в процессе. А редкое действие с небольшими последствиями обычно не заслуживает отдельного сценария.

Так аналитик помог Ирине увидеть разницу между расплывчатой задачей «бот отвечает на вопросы о ремонте» и конкретными действиями: сразу сообщать условия диагностики, собирать данные для записи, подтверждать полученную заявку. По журналу было видно, что первая задача повторяется часто, вторая требует аккуратного переноса, а точный срок ремонта должен по-прежнему определять мастер. Теперь можно было проектировать не универсального консультанта, а ограниченный сценарий с понятной границей ответственности.

Как провести собственную проверку

Для аудита не нужна сложная система учёта. Достаточно последовательно пройти путь последних обращений, не выбирая только удобные примеры.

Сначала определите период и объём. Для малого бизнеса подойдёт последний рабочий период, в котором набралось примерно двадцать-тридцать обращений, или последние десять рабочих дней. Если обращений мало, возьмите все за месяц. Если их слишком много, выберите заранее понятный отрезок — например, каждое третье обращение из хронологического списка. Не собирайте только истории, которые запомнились как неудачные.

Затем договоритесь, что считать одним обращением. Повторный вопрос по той же записи, звонок после сообщения и переход между каналами нужно отмечать так, чтобы не посчитать одного человека несколько раз. Иначе бизнес будет сравнивать не заявки, а количество касаний. Касания тоже можно считать, но отдельно.

Для каждого случая восстановите последовательность: первое сообщение, первый содержательный ответ, уточнения, перенос сведений, подтверждение и итог. Если история обрывается, поставьте пометку «исход неизвестен». Не заполняйте пробел догадкой. Важно учитывать не только успехи и отказы, но и незавершённые случаи: они показывают, где процесс теряет ясность.

Теперь отметьте повторяющиеся действия: одинаковые ответы, ручное копирование, поиск сведений у коллеги, повторный звонок, исправление записи. Время ожидания и время работы сотрудника записывайте отдельно. Ошибки описывайте конкретно: «не указан день», «номер введён с ошибкой», «сообщение не увидели до закрытия». Формулировка «плохо обработали» ничего не объясняет и легко превращается в обвинение сотрудника.

Наконец, выберите один главный показатель и несколько контрольных. Главный должен соответствовать найденному узкому месту. Для «Спицы» им стало время до содержательного ответа в рабочее время: оно было заметно по перепискам и напрямую связано с жалобой Ирины на ожидание. В качестве контрольных показателей мастерская решила отслеживать долю записей без ошибок и число уникальных обращений во всех каналах. Первый покажет, не растёт ли количество исправлений, второй поможет заметить, не перешли ли клиенты просто из одного канала в другой.

Не пытайтесь измерять всё сразу. Если главная проблема — перенос записи, считайте, сколько записей пришлось повторно вводить или исправлять. Если дело в долгом ожидании, измеряйте время до реального ответа, а не до приветствия. Если часто пропускают звонки, фиксируйте число пропусков и долю случаев, когда удалось связаться с человеком. Один ясный показатель полезнее десяти неясных.

Исходная точка до запуска

Чтобы после запуска понять, что изменилось, недостаточно сохранить общий вывод «раньше отвечали медленно». Нужно заранее договориться об исходном значении, периоде наблюдения и точном определении показателя.

У «Спицы» исходным значением для главной метрики стала медиана — двадцать восемь минут до содержательного ответа по девятнадцати обращениям, начатым в рабочее время. Дополнительно зафиксировали, что в пяти случаях из девятнадцати ответ занял больше часа. Период сбора — десять рабочих дней в текущем сезоне.

После запуска мастерская должна считать так же: учитывать те же часы работы и каналы, использовать то же определение содержательного ответа и брать сопоставимый период. Если бот будет работать только в сообщениях сообщества, нельзя судить о результате лишь по этому каналу и забывать про телефон. Иначе может показаться, что ожидание сократилось, хотя клиенты просто начали звонить. Общее число уникальных обращений во всех каналах поможет заметить такой сдвиг.

Сравнивать разные сезоны тоже рискованно. В жаркий период велосипеды ремонтируют чаще, чем зимой; очередь, состав команды и доля срочных работ могут быть совсем другими.

Если сопоставимого периода нет, зафиксируйте различия и не приписывайте боту все перемены. Количество заявок могло вырасти из-за сезона, рекламы или изменения расписания мастерской.

До запуска можно сформулировать ожидаемый эффект, но не выдавать его за доказанный. Для «Спицы» ожидание такое: часть типовых вопросов будет получать быстрый ответ, а данные для записи станут приходить в более удобном формате. Доказательство появится только после сравнения: сократилось ли ожидание, стало ли меньше исправлений, не прибавилось ли сообщений, которые Павлу приходится разбирать вручную. Обещать рост продаж только на основании найденных пауз нельзя.

Полезно заранее выбрать и защитный показатель — то, что не должно ухудшиться. Например, долю записей с ошибочной датой, количество обращений, не переданных сотруднику, или число клиентов, которым пришлось повторно объяснять свой запрос. Бот, который отвечает быстро, но собирает неполные сведения, улучшит одну цифру за счёт всего пути клиента.

Что чаще всего искажает расследование

Самая распространённая ошибка — выбирать только самые яркие истории. Случай, когда клиент ушёл к другой мастерской, важен, но не показывает, насколько часто возникает проблема. Сохраните его как отдельный эпизод и сравните с обычными обращениями.

Ещё одна ошибка — считать автоматическое приветствие полноценным ответом. Клиент, который спросил о сроках, не получил решения от сообщения «Спасибо, скоро ответим». Если измерять только первое касание, ожидание можно сократить на бумаге, ничего не изменив в пути человека.

Опасно считать каждое молчание потерей продажи. Возможно, клиенту уже помогли по телефону, он отложил решение или переписка оборвалась по другой причине. Пометка «исход неизвестен» точнее, чем выдуманная причина.

Не стоит и складывать все минуты, а потом называть результат экономией. Даже если сотрудник тратит время на типовые ответы, бот не обязательно снимет всю эту работу: часть клиентов задаст дополнительные вопросы, часть ситуаций потребует участия человека, а собранные данные придётся проверять. До запуска это только возможная экономия; после запуска её нужно измерить.

Наконец, аудит не должен превращаться в оценку сотрудника. Журнал показывает, как устроен процесс, а не насколько старателен Павел. Если сообщения приходят в разные места, расписание ведётся в двух форматах, а сотрудника одновременно отвлекают звонки, задержка может быть предсказуемым результатом самой системы. Цель проверки — найти условия, в которых ошибка повторяется, а не назначить виноватого.

Упражнение на ближайшие полчаса

Возьмите последние десять-пятнадцать обращений, если их немного, или последние двадцать-тридцать, если поток плотный. Для каждого запишите канал, время первого сообщения, время содержательного ответа, итог, ручные действия и возможную ошибку или потерю. Не заполняйте пробелы по памяти — поставьте пометку «неизвестно».

Затем отметьте действия, которые повторяются. Посчитайте, как часто они возникают, оцените, сколько времени или исправлений требуют, и спросите себя: можно ли заранее сформулировать верный ответ или правило? Если можно, это кандидат на автоматизацию. Если нельзя, но важно не потерять запрос, бот может собрать исходные сведения и передать их человеку. Если же проблема связана с несколькими журналами, сначала исправьте учёт.

Такая проверка пригодится не только мастерской. Небольшой интернет-магазин может обнаружить, что покупатели часто спрашивают о наличии товара, а остатки в переписке и учётной таблице расходятся. В этой ситуации бот не должен уверенно сообщать о наличии по уста-

ревшим данным. Сначала нужно определить, где хранится актуальный остаток. После этого бот сможет показывать его или собирать запрос на проверку.

Тот же механизм виден и в семейной переписке. Родители договариваются в общем чате, кто заберёт ребёнка с занятия, но один считает, что договорённость состоялась, а другой ждёт подтверждения. Бот здесь не всегда нужен: возможно, достаточно общего календаря и явного подтверждения ответственного. Автоматизация имеет смысл, если она снимает повторяющиеся напоминания, а не маскирует отсутствие договорённости.

Журнал последних обращений нужен не для того, чтобы доказать необходимость бота. Он помогает принять более полезное решение: автоматизировать конкретное действие, изменить процесс без бота или пока ничего не менять. В «Спице» обнаружили типовые вопросы, задержки и ручные переносы, но ещё предстояло выяснить, какую часть работы можно безопасно передать автоматике и кому она действительно поможет. Следующий шаг — определить, кто обращается, в каком канале и в какой момент его путь нуждается в поддержке.

Кому, где и в какой момент нужен бот

Бот может экономить владельцу время и одновременно раздражать клиента, если его проектируют только ради внутреннего удобства. Клиент заполняет длинную форму, сотрудник получает неполные данные, а владелец видит автоматизацию, которая не избавила ни от задержек, ни от повторных вопросов.

Разобрав недавние обращения и отметив, где теряются время и заявки, стоит задать следующий вопрос: кто именно сталкивается с каждым узким местом? Клиент, сотрудник и владелец видят одну и ту же операцию по-разному. Поэтому сценарий бота нужно проверять не только на соответствие задаче, но и на удобство для каждого участника.

Три пользователя — три критерия

Клиенту нужен понятный результат и минимум усилий. Он может хотеть узнать цену, выбрать время, проверить статус или сообщить об изменениях. Бот удобен, если помогает сделать это быстрее и не заставляет разбираться в устройстве компании.

Сотруднику важно получить то, с чем можно работать: заявку с нужными данными, запись в расписании, ясный статус или историю договоренностей. Если бот собирает сведения в одном месте, а сотруднику приходится вручную переносить их в другую систему, автоматизация не снимает работу, а лишь перемещает ее.

Владельцу нужны управляемость и измеримый эффект. Например, он хочет сократить время ответа, уменьшить число пропущенных обращений или снизить количество однотипных звонков. Сам факт запуска бота ничего из этого не доказывает.

Если смешать эти критерии, возникают предсказуемые сбои. Владелец хочет собирать как можно больше данных — и бот задает клиенту десять вопросов, прежде чем показать свободное время. Сотруднику нужна полная информация о ремонте — и клиенту предлагают подробно описать неисправность, хотя на экране телефона это неудобно. Клиенту достаточно узнать, готов ли заказ, а бот предлагает заново пройти все меню услуги.

Перед автоматизацией сформулируйте три результата отдельно:

Клиент должен суметь...

Сотрудник должен получить...

Владелец должен увидеть...

Если все три фразы описывают одно и то же, проверьте, не подменили ли вы потребность пользователя внутренней целью бизнеса. Клиенту незачем знать, какую таблицу заполняет сотрудник. Сотруднику не обязательно видеть рекламные предложения для клиента. Владелец нужен показатели и информация об исключениях, а не копия каждого шага интерфейса.

Клиент проходит услугу по времени

Потребности клиента меняются в зависимости от этапа. Один сценарий «ответить на вопросы» редко одинаково полезен до, во время и после услуги.

До услуги человек решает, подходит ли ему предложение и что делать дальше. Для записи в студию это могут быть цена, свободное время, длительность процедуры и правила отмены. Для ремонта техники — список принимаемых устройств, порядок диагностики, примерные сроки и способы передать устройство. Для доставки — дата, ограничения по адресу и порядок переноса.

На этом этапе бот должен уменьшать неопределенность, а не просто выдавать рекламный текст. Если цена зависит от состояния устройства или объема работ, не стоит изображать точный расчет, когда данных для него еще нет. Лучше объяснить, от чего зависит сумма, и предложить следующий шаг: описать задачу, приложить допустимые материалы или связаться со специалистом.

Во время услуги на первый план выходят статус и возможность что-то изменить. Клиенту нужно узнать, принята ли заявка, когда ждать мастера, можно ли перенести запись и что делать при задержке. Ответ «Ваша заявка зарегистрирована» полезен только тогда, когда понятно, что произойдет дальше и где получить обновление.

При доставке тот же принцип означает не просто повторить номер заказа, а предложить действие: уточнить интервал, проверить статус или сообщить об изменении адреса. Если автоматически изменить маршрут нельзя, бот должен сразу обозначить это и передать обращение сотруднику. Пустое обещание «Мы скоро ответим» лишь добавляет повод для повторного звонка.

После услуги клиенту могут понадобиться инструкция, документы, гарантийные условия, информация об оплате или способ сообщить о проблеме. Здесь пригодится короткий порядок действий, связанный с уже оказанной услугой. Если бот не знает, о каком заказе идет речь, не заставляйте человека угадывать формат ввода: предложите выбрать заказ, указать его номер или обратиться к сотруднику.

Разделение по этапам помогает не раздувать сценарий. Необязательно включать в один бот все возможные вопросы о компании. Если чаще всего спрашивают до записи, начните с информации и выбора времени. Если основная нагрузка возникает после покупки, проверьте статусы и инструкции. В первую очередь автоматизируйте тот этап, где часто повторяются запросы и для них есть понятное решение.

Сотруднику нужен не еще один чат

Внутренний маршрут бота оценивают по тому, что происходит с обращением дальше. Попадают ли данные в систему, где ведутся заявки? Видит ли сотрудник, что клиент уже сообщил? Понятно ли, кто будет отвечать и в какой срок?

В небольшом сервисе бот может передавать структурированное обращение в рабочую панель или другой инструмент, которым пользуется компания. Важна не конкретная система, а отсутствие лишнего ручного переноса. Если данные приходят в личный чат сотрудника, затем пересылаются коллеге и теряются при смене дежурства, новый канал создал уязвимость вместо порядка.

Сотруднику также нужно иметь возможность исправить неверно выбранную категорию, отметить исключение и передать вопрос тому, кто вправе его решить. Бот не должен закрывать путь к человеку только потому, что пользователь выбрал не тот пункт меню. Внутренний сценарий должен учитывать не только обычную заявку, но и возврат на доработку, отказ от услуги, повторное обращение и технический сбой.

Заранее определите, какие данные сотруднику необходимы. Например, обращение о переносе записи может содержать выбранную услугу, дату, контакт для связи и желаемое новое время. Если для решения нужны дополнительные сведения, сотрудник запросит их позже. Просить клиента сообщить их «на всякий случай» — значит усложнять сценарий без гарантии пользы.

Особенно внимательно проверьте случаи, когда заявку принимает один человек, работу выполняет другой, а за оплату отвечает третий. Бот не должен создавать видимость единого процесса, если у подразделений разные правила или нет общего статуса. Прежде чем обещать автоматическое обновление, выясните, откуда бот узнает о фактических изменениях. Если статус вносят вручную, назначьте ответственного и определите, когда именно он должен его обновить.

Владельцу нужны границы и измеримый эффект

Владелец смотрит на процесс целиком: сколько обращений поступает, где возникают задержки и какую повторяющуюся работу можно снять. Для оценки бота важны не все сообщения подряд, а показатели, связанные с исходной проблемой.

Если узким местом были вопросы о графике, отслеживайте, сколько пользователей получили ответ без помощи сотрудника и сколько все-таки пришлось передать ему. Если проблема заключалась в пропущенных заявках, важно знать, сколько обращений дошло до отправки и сколько получили дальнейшую обработку. Если автоматизировали запись, учитывайте не только число начатых сценариев, но и долю завершенных записей, отмен и переходов к человеку.

Одних показателей недостаточно. Бот может отвечать быстро, но отправлять людей не в тот раздел. Может собирать много заявок, но с неполными или неверными сведениями. Периодически просматривайте несколько завершенных и незавершенных маршрутов, соблюдая внутренние правила работы с данными. Так можно понять, где человек остановился и почему.

Владельцу нужно также определить границы автоматизации. Какие случаи бот решает сам? Чего он не обещает? Кто следит за расписанием, ценами и инструкциями? К кому поступает обращение при сбое? Без ясных ответов простая правка прайса или смена графика может запустить цепочку ошибок.

Канал выбирают там, где возникает задача

У каждого канала свои преимущества. Выбирайте его с учетом того, где клиент уже взаимодействует с компанией, какое действие ему нужно выполнить и доступен ли ему этот способ в конкретной ситуации.

Сообщество ВКонтакте подойдет, если аудитория уже находит компанию через ее страницу и привыкла писать в сообщения сообщества. Здесь удобно связать публикации, описание услуг и обращение в одном знакомом пространстве. Но сама страница не гарантирует, что клиент заметит нужный раздел, войдет в аккаунт или предпочтет переписку звонку. Если большинство обращений приходит через сайт или по телефону, не обязательно переносить их во ВКонтакте только потому, что там можно настроить бота.

Telegram уместен, если клиенты уже используют его для общения или если им удобно открыть бота по прямой ссылке, например из сообщения о записи. Этот канал может подойти для повторных запросов, уведомлений и проверки статуса. Но клиенту придется найти и открыть нужного бота, а часть аудитории может вообще не пользоваться Telegram. Ссылка упрощает переход, но не гарантирует, что человек выберет именно этот способ.

Сайт подойдет, если пользователь уже открыл страницу услуги и хочет завершить действие там же. Чат или форма позволяют не переключаться между площадками. Но человеку может быть трудно вернуться к диалогу позже, особенно если он закрыл страницу или сменил устройство. Продумайте, как он получит подтверждение и сможет продолжить обращение.

Телефон остается важен, если нужно быстро уточнить детали, у клиента нет удобного доступа к цифровому каналу или ситуация вызывает сильные эмоции. Звонок не всегда нужно заменять ботом. Иногда разумнее автоматизировать подготовку: предложить оставить контакт и коротко обозначить тему, чтобы сотрудник не начинал разговор с нуля.

Почта удобна для подробных итогов и документов, но обычно хуже подходит для быстрого выбора действия или проверки статуса. СМС может передать короткое уведомление или напоминание, однако для длинного диалога и выбора из множества вариантов это тесный формат. Канал не обязан выполнять все функции: о событии можно сообщить в одном месте, а сложный вопрос продолжить решать в другом, заранее объяснив переход.

У выбора есть и организационная сторона. Публичный канал может быть удобен клиенту, но не подходит для передачи сведений, которые не следует размещать в открытом поле. Если сценарий затрагивает персональные данные, собирайте только необходимое и используйте каналы и способы хранения, предусмотренные правилами компании и применимыми требованиями законодательства РФ, включая требования к обработке персональных данных.

Не просите отправлять чувствительную информацию в общий чат только потому, что так проще настроить бота.

Для проверки задайте себе простой вопрос: откуда человек приходит к задаче? Если он видит услугу на сайте и хочет записаться, оставьте короткий путь к записи там же. Если постоянный клиент открывает ссылку из сообщения о заказе, ему может быть удобнее бот в привычном мессенджере. Если важно подробно объяснить ситуацию, а не выбрать один из вариантов, сохраните возможность написать специалисту или позвонить.

Матрица маршрутов

Сведите три перспективы в одну рабочую схему. Для каждой роли укажите не площадку, которую хочется подключить, а место, где человек действительно готов решать свою задачу.

Клиент до услуги. Задача — понять условия, выбрать услугу или время. Канал — страница сообщества ВКонтакте, Telegram или сайт, в зависимости от того, откуда пришел человек. Момент — когда он сравнивает варианты или готов записаться. Запасной путь — телефон, форма или обращение к сотруднику.

Клиент во время услуги. Задача — проверить статус, изменить запись или получить инструкцию. Канал — тот, где оформлена услуга, либо ссылка из подтверждения. Момент — после события, повлиявшего на статус, или по запросу клиента. Запасной путь — связь с ответственным сотрудником.

Клиент после услуги. Задача — найти документы и правила или сообщить о проблеме. Канал — сообщение, сайт или почта, в зависимости от вопроса. Момент — после завершения заказа или при возникновении проблемы. Запасной путь — телефон или передача обращения специалисту.

Сотрудник. Задача — получить полное обращение и продолжить работу. Канал — рабочая система, панель заявок или утвержденный внутренний канал. Момент — после отправки заявки или срабатывания правила передачи. Запасной путь — назначенный коллега или руководитель смены.

Владелец. Задача — контролировать результат и замечать сбои. Канал — сводные показатели и выборочная проверка сценариев. Момент — по расписанию и после заметных изменений в процессе. Запасной путь — ручной разбор обращений и корректировка правил.

Матрица не должна превращаться в обещание, что все маршруты заработают одновременно. Ее задача — выявить разрывы. Например, клиенту предлагают обратиться к специалисту, но не сообщают часы его работы; сотруднику отправляют заявку, но не указывают, откуда она пришла; владелец видит число начатых диалогов, но не знает, сколько людей получили нужный результат.

Проверьте каждую строку по пяти вопросам. Понятно ли, какую задачу решает человек? Доступен ли ему выбранный канал? Подходит ли момент контакта? Получит ли он ясное подтверждение? Есть ли запасной путь, если автоматический сценарий не подходит?

Доступность проверяется на сбоях

Удобный сценарий не требует от человека цифровой сноровки. Пользователь может не знать, что нужно нажать кнопку под сообщением, не понять термин или случайно закрыть окно. Это не его ошибка, а обстоятельство, которое интерфейс должен учитывать.

Вместо команды «Введите код услуги в установленном формате» объясните, где найти код, или предложите выбрать услугу из короткого списка. Вместо просьбы подробно описать проблему дайте выбрать несколько понятных признаков и добавить комментарий. Не заставляйте вводить номер телефона в международном формате, если достаточно выбрать страну и набрать привычный номер.

На каждом шаге показывайте, что уже известно и что нужно сделать дальше. Не спрашивайте то, на что человек только что ответил. Дайте возможность вернуться на предыдущий

шаг, исправить ответ или начать заново. Кнопка «Связаться со специалистом» должна быть на виду, а не появляться после нескольких кругов меню.

Учитывайте обстоятельства, в которых человек открывает бота. Клиент может быть в дороге, пользоваться небольшим экраном, находиться в шумном месте или иметь нестабильное соединение. Длинные инструкции, большие блоки текста, обязательные голосовые сообщения или требование сразу приложить несколько файлов могут помешать ему выполнить задачу. Если объяснить проблему можно только голосом, человек, которому неудобно записывать или слушать такие сообщения, останется без подходящего маршрута.

Для регулярных действий выбирайте короткие и понятные шаги. При сложном выборе пояснения необходимы, но лучше разместить их рядом с соответствующим пунктом, а не выдавать длинную инструкцию в самом начале. Проверьте сценарий на реальном телефоне и попросите людей с разным опытом пройти его без подсказок автора. Если они ищут кнопку, не понимают формулировку или боятся нажать не туда, фраза «все и так очевидно» интерфейс не исправит.

Доступность означает и возможность отказаться от автоматического пути. Человек может не хотеть регистрироваться на новой площадке, не иметь нужного приложения или просто предпочитать телефон. Не считайте такой выбор сопротивлением цифровизации. Если альтернативный путь поддерживает процесс, покажите его до того, как клиент потратит время на неудачную попытку.

Когда сценарий должен уступить человеку

Автоматизация хорошо справляется с известными вариантами. Живой разговор нужен там, где приходится учитывать контекст, принимать нестандартное решение или поддерживать человека в напряженной ситуации.

Первый сигнал — сложный выбор. Если решение зависит от нескольких условий, которые нельзя надежно свести к короткому меню, бот может собрать исходные данные, но не должен изображать персональную консультацию. Например, у устройства может быть несколько неисправностей, а цена ремонта зависит от диагностики. Бот способен объяснить порядок действий и записать описание, но решение остается за специалистом.

Тревога и конфликт требуют другого темпа и тона. Жалобу на сорванный срок, спор об оплате или опасение за качество не стоит проводить через длинную цепочку стандартных вопросов. Бот может зафиксировать номер заказа и тему, сообщить, кому направлено обращение, и передать его человеку. Нейтрально подтвердить получение обращения уместно; обещать клиенту результат, чтобы автоматически его успокоить, — нет.

Нестандартный случай возникает, когда ни один пункт меню не подходит. Если в интерфейсе нет варианта «Другой вопрос», человек выберет что-нибудь наугад, и обращение может попасть не тому сотруднику. Предусмотрите возможность описать ситуацию своими словами и настройте передачу человеку, если система не распознала ответ или пользователь несколько раз возвращается к одному шагу.

Еще один сигнал — трудности с цифровым интерфейсом. Повторные ошибки, просьба связаться с человеком, незавершенный сценарий или долгая пауза после сложного вопроса не доказывают, что пользователь не способен разобраться. Но это признаки того, что автоматический путь не сработал. В такой ситуации нельзя снова и снова показывать те же инструкции.

Передача обращения человеку должна быть частью процесса. Сообщите, какие сведения будут ему переданы, и не заставляйте клиента повторять уже сообщенное. Обозначьте часы ответа и следующий шаг. Если сотрудник подключится не сразу, честно скажите, сколько придется ждать, а не обещайте мгновенную реакцию без оснований. В срочной ситуации, связанной с угрозой жизни или безопасности, обычный бизнес-бот не должен быть единственным источником помощи: направьте человека в соответствующие экстренные службы, в России — по номеру 112.

Выбор можно свести к нескольким развилкам.

Если запрос повторяется, ответ известен заранее, а ошибку легко исправить, автоматизируйте весь короткий маршрут.

Если запрос повторяется, но для решения не хватает данных, автоматизируйте сбор минимально необходимых сведений и передачу обращения сотруднику.

Если случай редкий, цена ошибки высока или решение зависит от личной оценки, оставьте человека главным, а боту поручите навигацию и подготовку обращения.

Если клиент не завершает сценарий или выбирает неподходящий пункт, не добавляйте еще один уровень меню. Проверьте формулировки и сделайте передачу человеку доступной раньше.

Практика: опишите один маршрут

Выберите одну повторяющуюся задачу: например, запись на услугу, проверку статуса ремонта или перенос доставки. Не пытайтесь сразу охватить весь процесс компании. Заполните матрицу на основе реальных обращений, а не предположений о том, как «обычно» ведут себя клиенты.

Сначала сформулируйте результат для клиента одним предложением: например, «клиент выбирает свободное время и получает подтверждение». Затем опишите, что нужно сотруднику: «запись с выбранной услугой, временем и контактом попадает в рабочее расписание». Наконец, укажите, что будет проверять владелец: например, долю завершенных записей и причины перехода к сотруднику.

Теперь определите точку входа. Клиент пишет в сообщество ВКонтакте, переходит по ссылке в Telegram, открывает сайт или звонит? Не выбирайте несколько каналов только ради охвата. Начните с того, где уже возникает задача и где компания сможет поддерживать информацию в актуальном состоянии.

Проверьте три возможных сбоя. Что увидит человек, если свободных интервалов нет? Что произойдет, если он передумает или укажет неверные данные? Куда попадет обращение, если ни один вариант не подходит? Если на какой-то вопрос ответ звучит как «разберемся вручную», назначьте ответственного и определите порядок действий до запуска.

Затем попросите человека, не участвовавшего в проектировании, пройти сценарий с телефона. Не подсказывайте, куда нажимать. Посмотрите, где он останавливается, что перечитывает и какого шага ждет. Такая проверка полезнее обсуждения кнопок в команде: авторы уже знают, что имели в виду, а пользователь видит только написанное на экране.

В результате должна получиться не карта всех будущих возможностей, а один проверяемый маршрут с ясной аудиторией, каналом, моментом контакта и запасным путем. Если клиент, сотрудник и владелец получают каждый свой результат, бот становится частью процесса, а не дополнительной вывеской. Если хотя бы одному из них приходится обходить сценарий вручную, не спешите добавлять функции: сначала найдите разрыв в маршруте.

Следующий шаг — ограничить запуск так, чтобы проверить его на реальных обращениях без долгого проекта и лишних функций. Начать стоит с небольшого маршрута, результат которого заметен клиенту, сотруднику и владельцу.

Маленький запуск вместо вечного проекта

Когда уже понятно, какую задачу решает проект, для кого он предназначен и в какой момент человек к нему обращается, возникает новая ловушка: функций становится больше, а их пользу никто не проверяет. В первую версию хочется включить всё сразу — запись, оплату, напоминания и ответы на любые вопросы. Но пока не ясно, помогает ли бот с главной задачей, каждая дополнительная функция увеличивает сроки, расходы и число причин отложить запуск.

Представьте мастерскую, где записаться на услугу можно сообщением, звонком или при личном визите. К вечеру на столе лежит целый список пожеланий: показать цены, подобрать услугу, проверить расписание, взять предоплату, напомнить о визите, предложить сопутствующие работы, собрать отзыв. А на проект есть один свободный вечер и ограниченный бюджет. Что оставить, если цель — не впечатлить демонстрацией, а проверить, станет ли записываться проще?

Пилот — это ограниченный запуск первой версии, который проверяет одну гипотезу за определённое время и в пределах доступных ресурсов. Он не обязан решать все задачи мастерской. Но должен доводить пользователя до понятного результата, а владельцу проекта — давать основания решить, что делать дальше.

Один результат вместо целого мира

Начните не со списка функций, а с того, чем должен закончиться путь пользователя. В нашем примере результат первой версии — не «бот ответил на приветствие» и не «клиент нажал кнопку». Мастерская должна получить заявку с данными, необходимыми для подтверждения записи, а пользователь — понять, что произойдёт дальше.

Для пилота достаточно одной точки, одного привычного канала и одного распространённого вида работ. Не нужно сразу переносить в бот весь каталог услуг. Выберите частую и относительно стандартную услугу, на которой можно проверить повторяющуюся проблему. Например, сотрудники тратят время, чтобы уточнить вид работы, удобное время и контакт клиента.

Минимальный полезный сценарий может выглядеть так:

1. Пользователь выбирает вид работы.
2. Указывает предпочтительный день или временной интервал.
3. Оставляет имя и контакт для ответа.
4. Получает сообщение о том, что заявка принята и её проверят.

5. Администратор получает сведения и подтверждает подходящее время или предлагает другое.

Если бот не подключён к актуальному расписанию мастерской, он не должен обещать, что запись уже подтверждена. Фраза «Вы записаны на вторник» может привести к сбою, если это время уже занято. Надёжнее написать: «Заявка принята. Администратор проверит расписание и подтвердит время». Это звучит скромнее, зато не обещает того, чего система пока не умеет.

На старте сотруднику остаются задачи, где нужно принять решение, а не просто собрать стандартные сведения: проверить свободное время, оценить нестандартную работу, ответить на вопрос вне короткого перечня, подтвердить перенос или отмену. Это не слабое место бота, а заранее продуманная граница между автоматизацией и работой человека.

Пользователь должен иметь возможность выйти из сценария, если его запрос не подходит ни под один из предложенных вариантов. Кнопка или понятная команда для связи с админи-

стратором — часть минимального сценария, а не функция «на потом». Если человек застрял, бот не упростил запись, а создал дополнительный барьер.

Упражнение: определить полезный финал

Запишите результат одним предложением по формуле: «После взаимодействия с ботом пользователь получает..., а сотрудник получает...».

Для мастерской формулировка может быть такой: «Пользователь отправляет заявку на распространённую услугу и знает, когда ждать подтверждения; администратор получает вид работы, предпочтительное время и контакт, не запрашивая эти сведения по отдельности».

Проверьте себя тремя вопросами. Можно ли однозначно понять, достигнут ли результат? Ясно ли, какую работу больше не придётся повторять вручную? Понимает ли пользователь, чем заявка отличается от подтверждённой записи? Если хотя бы на один вопрос нет ответа, сценарий пока описывает функцию, а не пользу.

Частая ошибка — считать результатом сам факт общения с ботом. Например, отчёт показывает, что пользователи открывали меню, но неизвестно, получили ли сотрудники пригодные заявки. Другая ошибка — обещать полноценную автоматическую запись, хотя бот только собирает пожелания. В таком случае сначала стоит пересмотреть формулировку результата, а не добавлять новые кнопки.

Что вычеркнуть без сожаления

На этапе проектирования функция кажется мелочью: добавить кнопку, ветку сценария или поле. Но для неё нужны текст, настройка, проверка, обработка ошибок и человек, который будет обновлять информацию. Поэтому у каждой функции первой версии должен быть строгий пропускной пункт: помогает ли она проверить главную гипотезу?

Если вы хотите узнать, уменьшится ли число уточнений при записи, интеграция с календарём может не понадобиться. Бот собирает предпочтительное время, а администратор подтверждает его вручную. Но если гипотеза в том, что пользователь сможет сам выбрать и сразу забронировать свободный слот, календарь уже не украшение, а необходимая часть проверки. Тогда стоит сократить пилот в другом месте — например, оставить одну услугу и одну точку.

Решение зависит от того, какое предположение вы проверяете, а не от общего правила «интеграции всегда потом». То же касается оплаты. Чтобы проверить, собирает ли бот полные заявки, предоплата не нужна. Если же вы проверяете сценарий «записаться и оплатить без участия администратора», оплата может стать его центральной частью. Но тогда придётся отдельно проверить платёжный процесс и условия возврата.

В первой версии мастерской обычно можно отложить полный каталог услуг, программу лояльности, сбор отзывов, рекомендации дополнительных работ, сложные напоминания и ответы на широкий круг вопросов. Не потому, что они бесполезны, а потому, что не помогают проверить саму запись. Всё, что не нужно для основной гипотезы, может подождать следующего решения.

Не добавляйте свободный диалог только ради впечатления. Если пользователь выбирает один из трёх понятных вариантов, фиксированное меню обычно проще проверить и поддерживать. Свободный ввод нужен, если без него человек не может описать запрос. Для пилота достаточно собрать короткое описание и передать его администратору — не обязательно пытаться автоматически распознать все возможные виды работ.

Упражнение: провести ревизию функций

Возьмите каждое пожелание и закончите фразу: «Без этой функции мы не сможем проверить, ...». Если продолжение звучит убедительно, функция может быть необходима. Если же получается «так будет современнее», «пользователям может понравиться» или «у конкурентов есть», перенесите её в список возможных улучшений.

Затем оцените цену исключения. Что произойдёт, если этой возможности не будет три недели? Если пользователь всё ещё сможет выполнить основную задачу через администратора,

а гипотезу удастся проверить, функцию можно отложить. Но если без неё человек получит ложное подтверждение или не сможет завершить путь, исключать её нельзя.

Чтобы не раздувать проект, разделите функции на три группы: обязательные для безопасного и понятного сценария, необходимые для проверки основной гипотезы и желательные, но отложенные. В мастерской к первой группе относятся точное сообщение о статусе заявки и возможность связаться с сотрудником. Ко второй — поля, без которых нельзя оценить качество заявки. К третьей — оплата, отзывы и расширенный каталог, если пилот проверяет только сбор данных для записи.

Тексты, которые задают границы

Сценарий может быть коротким, но пользователю важно понимать, что происходит. Для первой версии достаточно нескольких точных формулировок.

После отправки заявки: «Заявка получена. Это ещё не подтверждение записи. Администратор проверит расписание и ответит в рабочее время».

Для нестандартного запроса: «Этот запрос требует уточнения. Передайте его администратору, чтобы согласовать детали».

О сроке ответа: «Администратор обычно отвечает в течение одного рабочего дня». Указывайте срок, который мастерская действительно готова соблюдать, а не тот, что выглядит привлекательнее.

Не обещайте круглосуточную помощь, мгновенный ответ или гарантированно свободное время, если за этими словами не стоит реальный процесс. Бот может быть доступен всегда, а сотрудник, который подтверждает запись, — только в рабочие часы. Пользователь должен понимать эту разницу.

Если форма собирает телефон или другие персональные данные, запрашивайте только то, что необходимо для записи. Заранее определите, кто увидит заявки, где они будут храниться и как долго понадобятся. При запуске в РФ порядок обработки данных должен соответствовать действующим требованиям. Не передавайте сведения в сервисы и хранилища, использование которых не проверил ответственный за проект.

Числа: ориентир, а не обещание

Фраза «бот должен экономить время» не помогает принять решение. Нужна измеримая цель, но она не гарантирует, что результат будет достигнут. Цель задаёт порог, по которому вы решите, есть ли основания продолжать.

Сначала зафиксируйте исходное положение. Просмотрите последние обращения и посчитайте, сколько из них сразу содержали нужные сведения, сколько раз приходилось задавать уточняющие вопросы и сколько времени занимала обработка. Если данных нет, несколько дней отмечайте эти показатели вручную до запуска. Иначе сравнение «до и после» будет основано на впечатлении.

Допустим, за две недели в мастерскую поступило 20 подходящих запросов. В семи заявках сразу были указаны все сведения, необходимые для подтверждения записи. Исходная доля полных заявок — 35 процентов. Это условный пример: реальные значения нужно взять из собственных обращений.

Цель пилота можно сформулировать так: «За три недели не менее 60 процентов отправленных через бота заявок на выбранную услугу содержат все сведения для подтверждения без дополнительного запроса». Для оценки результата заранее задайте минимальный объём наблюдений — например, не менее 20 отправленных заявок. Если поток обращений в мастерской небольшой, подстройте срок и порог под реальность, а не подгоняйте данные под красивый план.

Здесь важно различать три вещи. Цель — желаемое значение показателя. Наблюдение — фактические данные за выбранный период. Обещание — утверждение о будущем результате, которое нельзя выводить из самого факта постановки цели. Честнее сказать «проверим,

достигнет ли доля полных заявок 60 процентов», чем обещать, что «бот сократит переписку на 60 процентов».

Основной показатель должен отвечать на вопрос, ради которого запускается пилот. Если гипотеза в том, что сотрудники станут реже задавать одни и те же вопросы, считайте долю заявок, обработанных без уточнений. Заранее определите знаменатель: это все отправленные заявки по выбранной услуге, а не все пользователи, открывшие бот. Отдельно отмечайте, сколько людей начали сценарий, но не завершили его. Этот показатель поможет найти проблему, но не заменит основной.

Добавьте защитный показатель, чтобы улучшение одной цифры не скрывало вред. Например, число потерянных или неправильно понятых заявок не должно расти. Если пользователи чаще отправляют неполные заявки, а администратор тратит на обработку больше времени, чем раньше, высокий процент нажатий на кнопку не означает успеха.

Карточка пилота

Зафиксируйте решение в одном месте до настройки бота. Карточку можно заполнить кратко, но ничего важного не пропускать.

Задача: уменьшить число повторных уточнений при записи на выбранную услугу.

Аудитория: люди, которые хотят записаться на эту услугу в одной мастерской.

Основной сценарий: выбор вида работы и предпочтительного времени, ввод контакта, передача заявки администратору и понятное сообщение о её статусе.

Срок: три недели наблюдения после проверки сценария. Если заявок не хватит, можно один раз продлить пилот на заранее установленный срок, например на две недели. Не продлевайте его только потому, что результат пока не тот.

Ресурс: один человек отвечает за настройку и проверку; администратор обрабатывает заявки в установленное рабочее время; на поддержку бота отводится, например, до 15 минут в день. Это ориентир, а не норматив. Установите свой лимит — такой, которого сможете придерживаться.

Бюджет: сумма, которую можно потратить без нового согласования. Учтите настройку, возможную плату за сервис, сообщения, интеграции и поддержку. Если пилот не укладывается в предел без подключения нескольких платных систем, пересмотрите сценарий и исключите функции, не нужные для проверки.

Показатель успеха: доля заявок, пригодных к подтверждению без дополнительного сбора обязательных сведений. Заранее укажите целевое значение и минимальное количество заявок.

Условие остановки: признаки, по которым пилот завершится, будет доработан один раз или прекратится.

Полезно указать и ответственных за повседневные задачи. Кто меняет перечень услуг? Кто видит новые заявки? Кто разбирается, если сообщение не дошло? Кто проверяет показатели в конце недели? Если на каждый вопрос ответ «кто-нибудь», пилот не ограничен на практике, даже если сценарий короткий.

Ограничения защищают проект от незаметного удорожания. Один канал вместо запуска сразу во всех каналах. Одна услуга и одна точка вместо полного каталога. Передача заявок администратору вместо сложной синхронизации расписания — если для проверки гипотезы она не нужна. Настройка на уже доступном конструкторе без найма разработчика — если возможностей конструктора хватает. Но «без кода» не значит «без ответственности»: за тексты, права доступа, проверку заявок и поддержку всё равно должны отвечать конкретные люди.

Условие, при котором проект можно остановить

Если не назначить дату пересмотра, пилот легко превратится в вечный проект. Каждую неделю найдётся причина продлить его ещё немного: добавить услугу, переписать приветствие, дождаться более активного периода. Поэтому решение о продолжении лучше записать до запуска, пока результат ещё не влияет на самооценку команды.

Предусмотрите три исхода. Продолжить — если набран минимальный объём данных, цель достигнута, защитный показатель не ухудшился, а обслуживание укладывается в установленный лимит. Один раз скорректировать — если люди начинают сценарий, но регулярно останавливаются на одном и том же шаге или администратору неудобно работать с полученными данными. Остановить — если при достаточной видимости ботом не пользуются, качество заявок не растёт, ручной работы становится не меньше или расходы превышают ожидаемую экономию.

Некоторые сбои требуют немедленно приостановить пилот: бот вводит пользователя в заблуждение, теряет заявки или выдаёт неподтверждённое время за свободный слот. Сначала устраните риск, а затем решайте, продолжать ли испытание. Это не то же самое, что завершить пилот из-за слабого результата: безопасность и точность важны на любом этапе.

Условия пересмотра должны быть конкретными. «Если что-то пойдёт не так» — не критерий. А вот «если более чем в 10 процентах заявок возникает ошибка передачи или администратор не видит заявку, приостанавливаем бота до исправления» — уже проверяемое правило. Порог выбирают с учётом допустимого риска; для серьёзной ошибки он может быть нулевым.

Не меняйте целевой показатель задним числом. Если первая версия не достигла порога, это не повод объявлять успехом любое улучшение. Зафиксируйте, что сработало, где возник сбой и к какому решению приводят данные. Если решите внести поправки, меняйте один ключевой элемент: иначе не получится понять, что именно повлияло на результат. После этого проведите один ограниченный повторный тест или закройте пилот.

Такой итог не обесценивает работу. Если бот не помог, вы выяснили это за ограниченное время и не стали расширять сценарий, который не решает нужную задачу. Если польза появилась, теперь есть основания развивать проект — опираясь на результаты, а не на список пожеланий.

Перед запуском проверьте границы

У пилота есть одна задача, одна аудитория и один основной путь от начала до результата.

Пользователь понимает, что бот может сделать сам, а что должен подтвердить сотрудник.

Каждая обязательная функция нужна для безопасного и понятного сценария или для проверки гипотезы.

Для основного показателя известны исходное значение, способ подсчёта, период наблюдения и минимальный объём данных.

Назначены ответственные за заявки, поддержку и оценку результата.

Установлены пределы времени и бюджета, а также дата принятия решения.

Определено, что делать в случае успеха, исправимого сбоя или отсутствия пользы.

Если хотя бы один пункт остался без ответа, пилот пока не готов. Не обязательно откладывать запуск на месяцы: иногда достаточно одного вечера, чтобы выбрать сценарий, написать честные тексты, заполнить карточку и решить, какие задачи останутся человеку. Но готовность к проверке — ещё не готовность продукта к любым ситуациям.

Первая версия нужна не для того, чтобы показать все возможности сервиса. Её задача — проверить одну полезную гипотезу в понятных границах. Когда они зафиксированы, следующий шаг — разложить выбранный путь на конкретные сообщения, варианты выбора, возможные ошибки и передачу заявки сотруднику. Так появляется сценарий разговора, который можно настроить и проверить.

Сценарий как карта разговора

В разгар сезона Марина пишет в мастерскую «Спица»: «Здравствуйте. Тормоза почти не работают. Можно сегодня посмотреть? Сколько это будет стоить?» Павел отвечает за будущего бота, но пока не настраивает его в сервисе: он набросал на бумаге нейтральное приветствие, список услуг и часы работы. Однако Марине от этого не легче: она не понимает, как оформить заявку, можно ли узнать цену и что делать, если свободного времени сегодня не окажется.

Команда уже определила задачу: бот должен разгрузить сотрудников от повторяющихся вопросов и собирать заявки на ремонт, не обещая того, чего пока не умеет проверять автоматически. Для первой версии этого достаточно. Теперь нужно превратить задачу в маршрут, который приведёт человека к понятному результату и позволит изменить решение по пути.

Список реплик ещё не сценарий

Павел начинает привычным способом: записывает фразы, которые должен произносить бот. «Добро пожаловать в “Спицу”». «У нас можно отремонтировать велосипед». «Мы работаем с десяти до восьми». «Оставьте номер телефона». Каждая фраза по отдельности уместна, но вместе они не отвечают на главный вопрос: что Марине делать дальше?

Сценарий — не набор заранее написанных реплик, а карта возможных путей пользователя. Она ведёт от первого сообщения к конкретному результату и предусматривает ответвления: вопрос, исправление, отмену, обращение к сотруднику. Реплики наполняют карту, но не заменяют её.

Представьте, что человек впервые вошёл в мастерскую. На стойке лежит список услуг, на стене висит расписание, рядом стоят визитки. Но никто не объясняет, куда обратиться, чтобы записаться. Информация есть, пути нет. В боте так бывает, когда после приветствия пользователь получает длинный перечень возможностей или должен угадать команду, которую предусмотрел автор.

Павел проверяет бумажный черновик простым вопросом: если Марина напишет обычными словами, не нажимая кнопки, поймёт ли бот, чего она хочет? В первоначальном варианте — нет. После приветствия черновик предлагает нажать кнопку, а фраза «Можно сегодня и сколько стоит?» не подходит ни к одному из предусмотренных вариантов. В таком разговоре человек застревает в самом начале.

Ирина, владелица мастерской, замечает ещё одну проблему. Бот предлагает записаться, но не объясняет, что именно считается записью. Заглянуть в расписание автоматически он не может: свободные окна Павел и механики отмечают вручную. Если бот напишет «Вы записаны на сегодня», клиент приедет, а мастерская может оказаться занята. Если же бот просто отправит заявку, Марина не поймёт, подтвердили ли ей время.

Поэтому сначала команда определяет, к какому результату должен привести разговор. Для Марины это не просто нажатие кнопки или отправка данных. Заявка должна попасть в мастерскую, клиентка — понять, что время ещё не подтверждено, и знать, где и когда ждать ответа. Если позже появится актуальный календарь, сценарий изменится: бот предложит свободные интервалы и подтвердит выбранный. Но это уже другая версия продукта, а не повод изображать возможность, которой пока нет.

В «Спице» очерчивают границы первой версии: бот отвечает на несколько проверенных вопросов, собирает запрос на ремонт и передаёт его сотруднику. Самостоятельно назначать время или рассчитывать стоимость ремонта по описанию неисправности он не будет. В этом примере диагностика стоит 900 рублей, а цену работ механик называет после осмотра. До запуска Ирина должна подтвердить эту сумму. Если реальные условия другие, изменится и сообщение: в сценарии нельзя оставлять цифру, за которую команда не готова отвечать.

Маршрут начинается с намерения

Первое сообщение бота должно не только приветствовать, но и подсказывать, куда можно обратиться. Если Марина уже написала, что хочет узнать цену, приветствие не должно заставлять её повторять вопрос. Бот может сразу ответить на ту его часть, для которой у него есть проверенная информация, и предложить следующий шаг.

Например, начать можно так:

«Здравствуйте! Помогу оставить заявку на ремонт, отвечу на частые вопросы или передам разговор сотруднику. Что вам нужно?»

Кнопки: «Записаться на ремонт», «Задать вопрос», «Изменить или отменить заявку», «Связаться с сотрудником».

Четыре варианта на первом экране — не самоцель. Здесь они обозначают четыре причины, по которым человек начинает разговор. Если кнопок становится слишком много, варианты стоит сгруппировать. А если нужного действия среди них нет, пользователь должен иметь возможность написать своими словами.

В сообщении Марины уже есть два намерения: узнать цену и попробовать записаться на сегодня. Бот может обработать их по очереди. Сначала ответить на известный вопрос, затем предложить оставить заявку:

«Диагностика стоит 900 рублей. Стоимость ремонта мастер назовёт после осмотра. Если тормоза не удерживают велосипед, не выезжайте на нём до проверки. Хотите оставить заявку на сегодня? Мастерская сверит расписание и сообщит, есть ли свободное время».

Кнопки: «Оставить заявку», «Ещё вопрос», «Позвать сотрудника».

Бот не обещает, что Марина точно попадёт сегодня. Он сообщает лишь то, что мастерская действительно может сделать: принять запрос и проверить расписание. Если сотрудник найдёт свободное время, он подтвердит его отдельно. Такая формулировка не звучит так эффектно, как мгновенная запись, зато не создаёт ложных ожиданий.

На бумаге маршрут выглядит так.

Вход — приветствие и распознавание или выбор намерения.

Запись на ремонт — тип велосипеда, описание проблемы, желаемый день или интервал и способ связи.

Проверка — сводка введённых данных с возможностью подтвердить их, изменить или прекратить заполнение.

Передача — заявка уходит в мастерскую; бот сообщает, что время пока не подтверждено, и объясняет, что будет дальше.

Вопрос — бот отвечает на проверенный вопрос или передаёт его сотруднику, а затем предлагает продолжить, вернуться в меню или закончить разговор.

Изменение или отмена существующей заявки — бот уточняет, какую именно заявку нужно изменить или отменить, либо направляет клиента к сотруднику.

Выход — бот подводит итог и оставляет возможность начать заново или обратиться за помощью.

Это пока не техническая схема: здесь нет названий блоков конструктора, интеграций и сложных условий. Зато уже видны главное — действия человека, реакция бота и понятный результат каждого пути. К переносу маршрута в конструктор Павел вернётся позже, когда команда выберет подходящий инструмент и проверит, что не пропустила важные развилки.

Как устроена ветка записи

Марина нажимает «Оставить заявку». Теперь бот должен собрать сведения, которые понадобятся сотруднику для первого контакта. Если спросить слишком мало, администратору придётся заново выяснять самое важное. Если спрашивать обо всём подряд, клиент может бросить заполнение, а часть данных так и не пригодится.

В «Спице» выбирают четыре пункта: тип велосипеда, описание неисправности, желаемое время и способ связи. Тип поможет механику понять, с каким заказом он столкнётся. Описание

неисправности даст представление о срочности и поможет подготовиться к разговору. Желательный интервал позволит сотруднику сверить расписание, а способ связи — сообщить результат.

Первый вопрос: «Какой у вас велосипед?»

Кнопки предлагают обычные варианты: «Городской», «Горный», «Детский», «Другой». Если ни один не подходит, Марина может написать ответ сама. Не стоит требовать от клиента точной технической классификации, если мастерская всё равно сможет уточнить детали позже.

Следом бот спрашивает: «Что нужно проверить или починить? Можно выбрать вариант или написать своими словами».

Кнопки: «Тормоза», «Колёса», «Переключение передач», «Другое».

Свободный текст здесь особенно важен: поломку трудно полностью уложить в короткий список. Марина может написать: «Тормозная ручка проваливается, переднее колесо скрипит». Если бот не умеет надёжно разобрать такое описание, ему не нужно изображать специалиста. Достаточно сохранить текст для механика, не ставя диагноз.

Затем — вопрос о времени: «Когда вам удобно привезти велосипед? Выберите день или укажите подходящий интервал. Время подтвердит администратор».

В первой версии лучше не показывать список якобы свободных окон, если он не обновляется автоматически. Можно предложить варианты «Сегодня», «Завтра», «Другой день», а затем попросить уточнить интервал свободным текстом. Кнопка «Сегодня» означает только пожелание клиента, а не бронь — если мастерская не может подтвердить время сразу.

Последний вопрос — о связи: «Как вам удобнее получить ответ: здесь или по телефону?»

Если Марина выбирает звонок, бот просит номер и объясняет зачем: «Укажите номер, по которому администратор сможет связаться с вами и подтвердить время». Если она предпочитает продолжить в том же чате, дополнительный номер может не понадобиться. Не стоит собирать данные «на всякий случай»: каждое поле требует усилий от пользователя и обязывает мастерскую бережно обращаться с полученной информацией.

Когда всё готово, бот не отправляет заявку молча, а показывает её целиком:

«Проверьте, пожалуйста: горный велосипед; проблема с тормозами, ручка проваливается; удобно сегодня после 17:00; ответить здесь. Время пока не подтверждено. Всё верно?»

Кнопки: «Отправить заявку», «Изменить», «Не отправлять».

Такая проверка даёт Марине возможность заметить ошибку до того, как сведения уйдут в мастерскую. Она могла случайно выбрать «Завтра» вместо «Сегодня» или неверно указать время. На этом шаге можно исправить и то и другое.

После подтверждения бот отвечает:

«Заявка № 184 принята. Вы просили проверить, можно ли приехать сегодня после 17:00. Администратор сверит расписание и ответит здесь. Это сообщение ещё не подтверждает время».

Номер в примере условный. Если система не присваивает заявкам номера, бот не должен их выдумывать. Достаточно написать: «Заявка передана администратору». Номер полезен, только если по нему сотрудник сможет найти обращение, а клиент — сослаться на него при уточнении.

Кнопки, свободный текст и возврат

Кнопки удобны там, где набор вариантов известен: выбрать действие, подтвердить данные, перейти к сотруднику. Они помогают избежать ошибок и показывают, что бот умеет делать. Но если оставить человеку только кнопки, разговор начинает напоминать анкету с закрытыми дверями. Не получится описать необычную неисправность, сообщить, что планы изменились, или задать короткий вопрос.

Свободный текст нужен там, где человеку проще сформулировать ситуацию самому. При этом команда должна заранее решить, что бот будет делать с сообщением: сохранит его как

описание, попробует найти ответ среди нескольких известных вопросов или передаст сотруднику. Если ответа нет, бот не должен угадывать.

Для первой версии «Спицы» выбирают простое правило. Кнопки ведут по основным маршрутам, а текст можно вводить в полях описания проблемы и уточнения времени. Если пользователь пишет что-то вне ожидаемого поля, бот распознаёт лишь несколько проверенных запросов — например, о цене диагностики и графике работы. Остальные вопросы он предлагает передать администратору.

Марина: «А сколько примерно займёт ремонт?»

Бот: «Срок зависит от неисправности и загрузки мастерской. После осмотра администратор или мастер сообщит, когда можно будет забрать велосипед. Продолжить оформление заявки?»

Кнопки: «Продолжить», «Спросить сотрудника», «Вернуться в меню».

Формулировка зависит от того, что мастерская может подтвердить. Ответ «обычно за час» не подойдёт, если в сезон ремонт часто занимает несколько дней. Бот должен опираться на проверенную информацию, а не на догадку, удобную для сценария.

Возможность вернуться на предыдущий шаг — не мелочь интерфейса, а способ снизить цену ошибки. Если на экране проверки Марина заметит, что выбрала не тот тип велосипеда, ей нужно изменить только этот пункт, а не проходить анкету заново. Если она ввела неверное время — вернуться к выбору времени. Если передумала на середине разговора — остановить заполнение.

В сценарии «Спицы» это предусмотрено заранее. Кнопка «Изменить» на экране проверки открывает выбор поля: тип велосипеда, описание, время или способ связи. После исправления бот снова показывает заявку целиком, чтобы Марина подтвердила обновлённые данные. Кнопка «Назад» на шаге описания неисправности возвращает к выбору типа велосипеда, сохраняя остальные ответы. А переход «В меню» возвращает к выбору действия. Если заявка ещё не отправлена, бот уточняет, хочет ли Марина прекратить заполнение.

Иногда вместо общей кнопки «Назад» полезнее дать конкретную: «Изменить день» или «Исправить описание». Название должно подсказывать, что произойдёт дальше. Кнопка «Далее» без пояснений заставляет гадать, особенно если перед пользователем несколько вариантов.

Свободный текст может прервать текущий шаг. Например, пока бот спрашивает, какой у Марины велосипед, она пишет: «Я передумала, не сегодня». Сценарий должен заметить намерение остановиться, а не принять эту фразу за ответ. Для первой версии не нужно обещать понимание любых формулировок. Достаточно распознавать несколько заранее проверенных команд, а в сомнительном случае уточнять:

«Вы хотите изменить день приезда или прекратить заполнение заявки?»

Кнопки: «Изменить день», «Прекратить заполнение», «Остаться на этом шаге».

Уточнение лучше неверного действия. Но если после каждого сообщения бот спрашивает «Что вы имели в виду?», он лишь перекладывает работу на клиента. Там, где варианты можно сформулировать ясно, нужно предложить их. Если намерение всё же неоднозначно — задать один вопрос и оставить возможность обратиться к сотруднику.

Вопрос и отмена — отдельные маршруты

У вопроса должен быть собственный путь и понятная точка возврата. Если Марина во время записи спрашивает о цене, бот отвечает проверенной информацией и предлагает продолжить с того места, где разговор прервался. Ему не следует забывать уже введённое описание неисправности и отправлять клиентку в начало.

«Диагностика стоит 900 рублей; стоимость ремонта мастер определит после осмотра. Продолжить заявку с выбора удобного времени?»

Такой переход показывает, что бот удерживает контекст разговора, а не перескакивает между несвязанными сообщениями. Если система не умеет сохранять текущий шаг, лучше честно предложить начать заново, чем создавать впечатление, будто данные остались в памяти.

Для незнакомого вопроса нужен другой ответ:

«Я пока не могу ответить на этот вопрос без уточнения. Передать его администратору?»

Кнопки: «Да, передать», «Вернуться к записи», «В меню».

Если сотрудник отвечает только в рабочее время, бот должен объяснить, когда ждать ответа. Не стоит обещать «сейчас подключим специалиста», если никто не дежурит. До запуска команде нужно определить срок обработки обращений или, если точный срок пока неизвестен, обойтись без конкретного обещания: «Администратор ответит после проверки сообщения в рабочее время».

С отменой тоже нужна ясность: пользователь может иметь в виду разные вещи. Марина может хотеть прекратить заполнение новой заявки, изменить уже отправленный запрос или отменить согласованный визит. Для каждого случая нужен свой маршрут.

Для незавершённой заявки бот пишет: «Заявка ещё не отправлена. Прекратить заполнение?»

Кнопки: «Да, прекратить», «Продолжить».

Если Марина подтверждает, бот отвечает: «Запрос не отправлен. Если захотите записаться позже, выберите “Записаться на ремонт”». Здесь нельзя писать «Ваша запись отменена»: записи ещё не было.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

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

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