ГЛАВА 1

Почему производительность — это пользовательский опыт

> Следует начинать с пользовательского опыта и уже от этого двигаться к технологиям. Нельзя сначала создать технологию, а затем думать о том, как ее продавать.
>
> — *Стив Джобс (Steve Jobs), 1997* (https://oreil.ly/_HYoc)

Эти слова Стива Джобса легли в основу данной книги: великие технологии начинаются с пристального внимания к пользовательскому опыту (user experience, UX). В мире веб-разработки *производительность* (performance) — один из важнейших аспектов UX, который, впрочем, нередко недооценивают. С веб-сайтом, который быстро загружается и моментально реагирует на действия пользователя, легко и приятно работать, а медленный и «задумчивый» вызывает чувство досады. Мне довелось много лет проработать в команде Google Chrome, и более десятка из них я посвятил тому, чтобы интернет стал быстрее и удобнее. Я занимался оптимизацией фронтенда, сгенерированного ИИ, стараясь сохранить на прежнем уровне отзывчивость и скорость веб-приложений, постепенно усложнявшихся за счет передовых технологий. В этой книге — моем сольном проекте — я поделюсь с вами мыслями и знаниями, которые я вынес из работы над повышением производительности (в том числе и над системой метрик Core Web Vitals, CWV), а также реальных UX-исследований и аудитов.

Работа с приложением или сайтом похожа на поездку по городу: ремонт дорог, светофоры и пробки портят все удовольствие. Так и задержки на сайте: неработающие кнопки, пустой экран при долгой загрузке, рывки при прокрутке — все это препятствия на пути, встретив которые люди испытывают стресс, теряют доверие и нередко уходят. Впечатление от сайта формируется за секунды. При этом, согласно исследованиям, большинство пользователей не возвращаются на сайты, которые при первом посещении показались им «глючными» или же слишком медленными.

Мнение о сайте начинает формироваться с момента перехода на него, а значит, *скорость* — это характеристика, которую человек отметит прежде всего. Изучая

взаимодействия людей и компьютеров (https://oreil.ly/BE8Zp), ученые заметили: любые, даже самые небольшие, задержки выбивают из потока, вызывая чувство досады. Отсюда следует золотое правило UX (https://oreil.ly/yW5Pr): люди обычно не замечают задержек в пределах 0,1 секунды, склонны мириться с секундным ожиданием, но спустя несколько секунд приходит острое осознание, что их заставляют ждать. Прождав десяток секунд, многие теряют терпение и отказываются от выполнения задачи.

Я написал эту книгу в расчете на разработчиков с уровнем подготовки от среднего до продвинутого, которые, несмотря на умение создавать функции и писать код, не в полной мере осознают, что производительность — фундамент UX. Здесь понятным, но профессиональным языком я объясню научные основы и принципы, которые связывают UX и скорость работы сайта, приведу ссылки на научные статьи и реальные примеры, которые покажут, как именно она влияет на удовлетворенность пользователей, их вовлеченность и, в конечном счете, успех бизнеса. Ведя рассказ о важных метриках и их значениях, я постараюсь объяснить вам, что производительность — залог хорошего UX, и показать, как оценивать ее на практике.

Сначала мы поговорим о том, что делает скорость столь важной для человека, его восприятия, о CWV и других показателях и моделях, которые помогают измерять производительность. Я расскажу о создании сайтов и приложений, способных работать на старых и маломощных устройствах при самых неблагоприятных сетевых условиях, упомяну о RAIL, Lighthouse, CrUX и других фреймворках и инструментах, незаменимых при анализе производительности.

Не фокусируясь на низкоуровневых методах оптимизации, я буду в основном говорить о принципах, метриках и результатах — о том, как быстрые сайты делают пользователей довольными.

Все остальные главы книги построены на этом фундаменте. В части II мы обсудим технические детали рендеринга в браузере, а также особые методы оптимизации CWV, в части III — проблемы, связанные с JavaScript, и способы их решения, в части IV — оптимизацию под часто применяемые сторонние скрипты. И наконец, в части V закрепим эти концепции: я приведу реальные примеры улучшения сайтов из разных предметных областей и дам полезные советы, которые помогут вам выстроить в компании культуру, где производительность — главный приоритет.

На протяжении всей книги я буду ссылаться на исследования (Google и других компаний), подтверждающие мои советы, и делиться с вами идеями, появившимися у меня за время работы в команде разработчиков Chrome. И я надеюсь, что, прочитав эту книгу, вы убедитесь в том, что производительность — это и есть UX, а высокая скорость сайта — ключ к сердцу пользователя.

Цена задержек: первое впечатление и удовлетворенность пользователя

Скорость — это не просто «приятная особенность» сайта. Она напрямую влияет на показатели бизнеса — отказы, конверсию и вовлеченность. Об этом говорят многочисленные отраслевые исследования. Согласно BBC (https://oreil.ly/EP3Dn, 2018 г.), с каждой секундой загрузки с сайта уходит до 10 % пользователей, и это довольно много: за несколько секунд можно лишиться половины целевой аудитории и существенно ухудшить конверсию. Ускорив загрузку сайта всего на 0,85 секунды, Walmart повысил конверсию на 2 % (https://oreil.ly/KWOxQ), а COOK — на целых 7 %. Эти примеры из реальной жизни подтверждают неразрывную связь производительности с успехом бизнеса и лояльностью пользователей.

Скорость — мгновенно оцениваемый показатель — сильно влияет на первое впечатление от сайта. Быстрый и плавный интерфейс кажется надежным и профессиональным, а медленный и подвисающий — небрежным и не вызывающим доверия. В наши дни ожидания пользователей как никогда высоки: считается, что сайт должен быть быстрым и отзывчивым, практически как обычное приложение. Терпение — дефицитный ресурс: многие люди не склонны ждать дольше нескольких секунд, особенно на мобильных устройствах (https://oreil.ly/WTn2r). Задержки, кроме того, вызывают беспокойство: что, если сайт завис, а не просто обрабатывает нажатие на кнопку?

Скорость — это краеугольный камень UX. Примите это как данность. Повышение производительности — не просто техническая задача, решением которой можно похвастаться, но нечто, напрямую связанное с удовлетворенностью пользователей, их вовлеченностью и даже позитивным восприятием бренда. Усвоив такое мышление, вы сможете подбирать технологии, ориентируясь главным образом на потребности людей (быстрый отклик, минимальное ожидание).

Но как понять, что конкретный сайт достаточно *быстрый*? И как определить, что он слишком *медленный*?

Согласно классическому правилу, которое сформулировал Роберт Б. Миллер (Robert B. Miller) — ученый, изучавший проблему времени отклика еще в 1968 году (https://oreil.ly/Pda59), — а в дальнейшем популяризировал Якоб Нильсен (Jakob Nielsen) (https://oreil.ly/lFJWP), задержки подразделяются на несколько уровней:

0,1 секунды и меньше

Операции, выполняемые за столь малый промежуток времени, воспринимаются как мгновенные: вам кажется, например, что от нажатия на кнопку и до реакции на него нет никакой задержки. Возникает иллюзия непосредственной работы с сайтом и полного контроля над интерфейсом.

Около одной секунды

Задержки в пределах одной секунды достаточно долгие, чтобы начать их чувствовать. Это примерное пороговое значение, до которого течение мысли

не прерывается, так как задержка воспринимается как естественная пауза. Однако еще чуть-чуть, и может показаться, что интерфейс занят чем-то другим.

От двух до девяти секунд

При паузах в несколько секунд сайт начинает действовать на нервы. Согласно исследованиям (https://oreil.ly/Kqfmj), чем выше время ожидания, тем ниже удовлетворенность. Конечно, многое зависит от контекста: к примеру, люди понимают, что видео обычно загружается дольше, чем выдаются результаты поиска. Однако в общем случае необходимо выводить на экран индикатор прогресса или какое-то сообщение, чтобы показать пользователю, что сайт выполняет его команду.

Десять секунд и более

Прождав так долго и не увидев реакции, большинство людей пытаются выполнить те же действия как-то иначе, а то и просто уходят с сайта. Но даже оставшись, они теряют фокус: переключаются на другую задачу или иным образом отвлекаются, а это значительно ухудшает пользовательский опыт.

Психология ожидания

Поговорим о том, как все эти пороговые уровни связаны с эмоциональными и когнитивными реакциями. Почему задержки свыше секунды так сильно раздражают людей? Возможно, вы уже знакомы с понятием «*поток*» (https://oreil.ly/D2WA-)[1] — состоянием сосредоточенности, при котором пользователь полностью погружен в выполняемую задачу. Если прервать поток, то человеку придется вспоминать, чем он занимался, и бороться с желанием переключить внимание на что-то другое.

Терпимость пользователя во многом зависит от контекста: срочная задача, например поиск нужной информации, делает любую задержку невыносимой, тогда как обычный просмотр сайтов вряд ли будет испытывать терпение пользователя. Играют роль и личностные качества. Поэтому нельзя назвать какое-то «идеальное», единое для всех количество секунд. Мы просто должны как можно сильнее ускорить работу сайта, не попадая в диапазоны, в которых задержки расстраивают большинство людей.

Задержки — это совсем не мелкие неудобства: они не только негативно влияют на UX, но могут приводить к ошибкам, повторным отправкам данных или переходам не на ту страницу. Стоит упомянуть и о «яростных кликах» (rage clicks) — когда, не видя ответной реакции интерфейса, пользователь вновь и вновь жмет на кнопку. По данным Akamai UX (https://oreil.ly/iPpls), так часто бывает, когда страница *выглядит* загруженной, а на деле не готова к работе. Более того, многие пользователи начинают взаимодействовать со страницей *до того,* как она станет полностью интерактивной. В более чем 30 % изученных случаев страницы все

[1] https://ru.wikipedia.org/wiki/Поток_(психология).

еще оставались непригодными для использования даже после срабатывания события `onload`. Около 15 % пользователей пытались совершить клик в промежутке между моментом визуальной отрисовки страницы и моментом, когда она стала интерактивной. Когда же сайт, наконец, загрузится и реально начнет функционировать, он может отработать повторные клики, и человек, который, например, несколько раз нажал на кнопку «Купить», фактически оформит не один, а несколько заказов.

Не менее проблематична и нестабильность макета: когда при медленной загрузке рекламы или картинок контент внезапно перемещается по экрану, нетерпеливый пользователь может нажать совсем не на ту кнопку или ссылку, на какую хотел.

Медленная работа сайта не ускользнет от внимания пользователей, и многие отреагируют на нее: кто-то будет яростно кликать, кто-то уйдет с сайта, а кто-то потеряет к нему интерес. Вот почему мы должны тщательно измерять и оптимизировать производительность.

Оценка UX-производительности: ключевые метрики

UX — это история, а не отдельное событие: нельзя описать весь процесс загрузки каким-то одним показателем. Создатели сайтов традиционно измеряют время до события `onload` (полной загрузки всех ресурсов) или другие временны́е интервалы, однако этого мало, чтобы понять, что ощущают пользователи. К примеру, если технически для загрузки нужны 5 секунд, но уже через две на экране мы можем видеть какой-то контент, то *реальные ощущения* будут намного лучше, чем можно предположить по `onload`. Обратная ситуация: сайт загружается за пару секунд, но 10 секунд не реагирует на команды, поскольку выполняется огромный JavaScript.

В этом разделе мы лишь кратко поговорим о метриках производительности и обсудим их важность. Подробное рассмотрение этой темы и всех упомянутых здесь показателей нам предстоит в главе 2.

Раньше внимание разработчиков было сосредоточено в первую очередь на технических критериях, которые не всегда отражают реальный UX. Я же предлагаю иной подход: нужно ответить на несколько простых вопросов, которые задает себе пользователь во время и после загрузки страницы:

Она загружается?

Сервер еще обрабатывает запрос или же снова пропал интернет? С этого все начинается: щелкнув по ссылке или набрав URL, человек ждет реакции. Важно как можно скорее сделать хоть что-нибудь (хотя бы сменить цвет фона, наметить какой-то интерфейс), чтобы человек был уверен: загрузка уже началась.

Релевантные метрики:

- Время до первой отрисовки (first paint, FP)
- Время до получения первого байта от сервера (time to first byte from server)

Она полезна?

Когда на экране появится нечто, что можно прочитать, посмотреть или понажимать? Уже не просто рамки и фон, а текст, картинка или же кнопки?

Релевантные метрики:

- Время до отрисовки первого контента (first contentful paint, FCP)
- Время до отрисовки крупнейшего элемента контента (largest contentful paint, LCP)

С ней можно работать?

Когда страница будет *готова* без промедлений реагировать на команды, станет интерактивной? Иначе, с точки зрения UX, нельзя считать, что она «загрузилась». Мы часто пытаемся нажимать на кнопки или прокручивать контент, стоит им только появиться, однако нередко это лишь видимость: на самом деле JavaScript еще готовит страницу к работе, и это требует времени.

Релевантные метрики:

- Время до интерактивности (time to interactive, TTI)
- Время от взаимодействия до следующей отрисовки (interaction to next paint, INP)

На ней приятно находиться?

И наконец, когда все загрузилось и начало работать, мы начинаем оценивать качество и стабильность: насколько плавно идет прокрутка, не подвисают ли анимации, не возникают ли сбои? Страница должна исправно работать, поскольку заминки или, напротив, внезапные рывки сбивают с толку и портят впечатление.

Релевантные метрики:

- Частота кадров
- Совокупное смещение макета (cumulative layout shift, CLS)

Производительность для всех: устройства и сетевые условия

Не все посетители сайта находятся в одинаковых условиях. В реальном мире производительность сильно зависит от устройств и сетей: у разработчика с мощным компьютером и оптоволоконной связью страница может летать, а у обычного человека с бюджетным Android и 3G — с трудом открываться. Поэтому если вы действительно заботитесь о UX, следует думать о равенстве: сайты должны полноценно работать у большинства посетителей (и чем их больше, тем лучше), а не только у обладателей новейших устройств.

Начнем разговор со сравнения телефонов с настольными ПК. У первых (особенно не самых дорогих) совсем другие, не слишком мощные процессоры, нередко меньше объемы памяти. К тому же и эти ресурсы бывают доступны не полностью: заняты выполнением фоновых задач или же ограничены для экономии заряда. Так что тяжелая веб-страница для них — задача более сложная, чем для настольного ПК. Большинство людей пользуются совсем не последней моделью iPhone: по данным на 2019 год (https://oreil.ly/h9GxP), больше 60 % смартфонов, используемых в мире, — это не флагманы Apple или Samsung, а устройства средней и низкой ценовой категории.

Возьмем, например, веб-приложение, загружающее JavaScript-код объемом 2 Мбайт. На новом настольном ПК с современным проводным интернетом он будет загружен за 2 секунды и выполнен за 200 мс, так что страница будет готова к работе всего через 2,2 секунды. А вот бюджетный смартфон потратит на ту же задачу 12 секунд: 8–10 из них уйдут на загрузку через 3G (или не очень надежную 4G) и еще две на выполнение. И это совсем другие ощущения. К тому же устройство может перегреться или внезапно понизить частоту процессора, что сделает ситуацию еще хуже. Вот так контекст, в котором находится пользователь, резко сужает окно производительности (диапазон условий приемлемой работы сайта).

Необходимо думать и о тех пользователях, которые находятся в «*хвосте*» распределения. Где-то за линией 90-го процентиля есть люди с устройствами пятилетней давности и плохим интернетом. И даже если у «среднего» посетителя все в порядке, *кому-то* приходится ждать по 10 секунд, и это негативный UX у немалой части аудитории. Поэтому я рекомендую рассчитывать бюджет производительности с оглядкой именно на такие крайние случаи (подробнее — в части II этой книги).

В мобильных сетях нередко бывают большие задержки (до сотен миллисекунд), что сильно замедляет загрузку, особенно если требуется серия запросов к серверу или необходимые ресурсы не оптимизированы. Производительность сайта упирается в фактическую скорость работы сети, что особенно чувствительно для больших изображений и видео.

Согласно некоторым исследованиям, владельцы мобильных устройств довольно терпеливы (https://oreil.ly/uV_8W): многие соглашаются ждать по 6–10 секунд, не покидая страницу. Возможно, они привыкли к тому, что в мобильном мире все медленнее. Однако это не означает, что 6 секунд — приемлемый результат: нет, это последний разумный предел. На деле, согласно оценкам Google (https://oreil.ly/Y2sFE), 53 % посещений мобильных сайтов заканчиваются уходом со страницы, если она загружается более 3 секунд. Так что терпение терпению рознь, и безопаснее все же оптимизировать сайт и делать его как можно более быстрым, в том числе для мобильных устройств.

В некоторых странах очень дорогой трафик, в других (главным образом в развивающихся) люди пользуются дешевыми устаревшими устройствами. В таких

условиях медленный и тяжелый сайт или веб-приложение не будут доступны по чисто финансовым соображениям. Стремясь обеспечить хороший UX для *всех,* необходимо думать и о подобных случаях. Планируйте бюджет производительности так, чтобы учесть ожидания всех людей, независимо от их устройств и скорости интернета. Условия могут быть очень разными, но хороший UX-инженер должен учитывать это разнообразие.

Дизайн с акцентом на производительность

В этом разделе представлены некоторые общие принципы проектирования и разработки с учетом производительности с самого начального этапа работы над сайтом. Советы здесь относятся скорее к общим подходам и принципам, а не к конкретным оптимизациям. Речь о том, чтобы закладывать производительность на этапах проектирования и архитектуры, а не не дорабатывать в последний момент.

Модель RAIL: производительность с ориентацией на пользователя

Прежде чем перейти к разговору о метриках CWV (см. главу 2), стоит упомянуть и об их предшественнице — модели RAIL (https://oreil.ly/XCHts), также разработанной командой Google Chrome и позволяющей взглянуть на производительность с точки зрения UX. Каждая буква в названии RAIL означает один из этапов работы веб-приложения и соответствующие требования к производительности: отклик (response), анимацию (animation), простой (idle) и загрузку (load). Достоинство RAIL заключается в том, что все эти цели производительности связаны с тем, что ощущают люди в разных сценариях. Поговорим о каждой из них.

Отклик

Сайт должен максимально быстро отвечать на действия пользователя (к примеру, нажатие на кнопку или запуск анимации). Согласно RAIL, на обработку события отводится 50 мс, на обновление страницы — 100, так чтобы человек не заметил задержки. Я же рекомендую не загружать обработчик сложной работой, которая может его замедлить. К примеру, при отправке формы достаточно сообщить о том, что данные приняты (визуальный фидбэк), и, может быть, отобразить какой-то индикатор, показывающий, что процесс идет. Все остальное должно выполняться в фоновом режиме. Тогда, даже не уложившись в 100 мс, вы по крайней мере оповестите пользователя о том, что нужно подождать.

Анимация

Прокрутка, анимация и переходы должны отображаться плавно, в идеале со скоростью 60 кадров в секунду. А это значит, что на каждый кадр отводится всего 16 мс. Рывки и заедания испортят впечатление даже от в целом быстрого сайта. Поэтому RAIL напоминает нам о том, что UX не ограничен одной лишь загрузкой.

Простой

Используйте время простоя с умом. Эта часть модели RAIL касается фоновой работы. Когда посетитель активно не взаимодействует с сайтом (к примеру, читает статью), мы можем выполнять второстепенные задачи (допустим, подгрузить очередную страницу). Однако делать это необходимо так, чтобы в любой момент, прервав фоновую работу, отреагировать на внезапные действия пользователя. Поэтому не загружайте основной поток непрерывной работой, а делите ее на части, чтобы сайт *оставался* отзывчивым: люди это оценят.

Загрузка

Старайтесь как можно быстрее доставить полезный контент. Согласно RAIL, мобильный сайт должен загружаться за 1 секунду. На практике это непросто, поэтому главный принцип — все нужно загрузить и вывести на экран как можно быстрее. Сайт должен стать полностью интерактивным за 2–5 секунд (в зависимости от контекста). Вот к чему надо стремиться. Каждая секунда задержки бьет по пользователю.

Все основные цели RAIL обобщены в табл. 1.1.

Таблица 1.1. Основные цели RAIL

Компонент RAIL	Целевое значение	Научное обоснование
Отклик	Время реакции на действие пользователя < 100 мс	На основе наблюдений Якоба Нильсена (https://oreil.ly/ATq-z) о реакциях, кажущихся мгновенными
Анимация	60 кадров в секунду (16 мс на кадр)	На основе исследований человеческого зрения и его способности сглаживать движения
Простой	Выполнение задач продолжительностью ≤50 мс	Позволяет браузеру своевременно реагировать на действия пользователя
Загрузка	Полезный контент загружается на среднее устройство за 1 с	На основе исследований Роберта Миллера (https://oreil.ly/ON8KJ) о состоянии потока и удержании внимания

В основе всех этих целей лежат научные исследования задержек и их влияния на ощущения людей и UX в целом.

RAIL — это ментальная модель, которая помогает инженерам, разработчикам и дизайнерам взглянуть на производительность с точки зрения пользователей.

Производительность как отправная точка проектирования

В начале работы над проектом следует определить *бюджет производительности* (performance budget), то есть ограничить «веса» страницы или приложения каким-то целевым уровнем. К примеру, можно потребовать, чтобы суммарный

объем JavaScript и CSS на главной странице не превышал 150 Кбайт, а суммарный объем сразу же загружаемых изображений — 100 Кбайт. Точные цифры здесь не важны (они определяются с учетом метрик, описанных в главе 2), но важен сам принцип: любые элементы, превышающие бюджет, должны быть пересмотрены. Смотрите на производительность как на одно из проектных требований, нарушить которое столь же недопустимо, как, например, разработать форму, не умещающуюся на стандартный экран или функционирующую только на самых мощных устройствах.

Постепенное улучшение (progressive enhancement) и *плавная деградация* (graceful degradation) — это стратегии, позволяющие обеспечить приемлемую производительность для всех пользователей. *Постепенное улучшение* заключается в том, что сначала на экране отображается упрощенный макет с ограниченной функциональностью (в расчете на устаревший браузер или же медленное устройство), который затем, по возможности, усложняется и расширяется. Вторая стратегия — *плавная деградация* — обратный вариант: первоначально сайт загружается в виде, рассчитанном на современный мощный браузер, но при необходимости (на слабом устройстве) становится менее изысканным, сохраняя при этом функциональность, пусть и с меньшим количеством возможностей. Таким образом, каждый получает хоть что-то. С точки зрения UX простой, но быстрый интерфейс лучше эффектного, но медленного.

Допустим, вы проектируете страницу поиска для сайта турагентства. В ответ на запрос клиент должен мгновенно увидеть несколько первых результатов (количество определяется скоростью загрузки). При этом картинки должны подгружаться по мере необходимости, начиная с тех, что сразу выводятся на экран, а карты и прочие второстепенные элементы — и вовсе заменяться заглушками. Определите, какая информация необходима *в первую очередь*, и загружайте ее как можно быстрее. Все прочее может подождать.

Обсуждайте производительность на проектных совещаниях. Рассматривая новые функции, оценивайте их влияние на время загрузки или интерактивность, и если предлагается, например, добавить скрипт объемом около мегабайта, сразу ищите альтернативы: возможно, те же задачи можно решить на стороне сервера или же после загрузки. Это вопрос не столько техники, сколько культуры: производительность должна быть первым критерием для принятия решений на любых уровнях. (Подробно о культуре производительности мы поговорим в главе 11.)

Взяв на вооружение эти принципы и образ мышления, вы сможете строить быстрые и приятные в работе сайты. Мне бы хотелось научить вас не просто измерениям и диагностике, но разработке с проактивным отношением к скорости. Производительность — это не дополнительный технический аспект, это философия дизайна. Приняв ее, вы сможете действовать более разумно с самого начала проектирования.