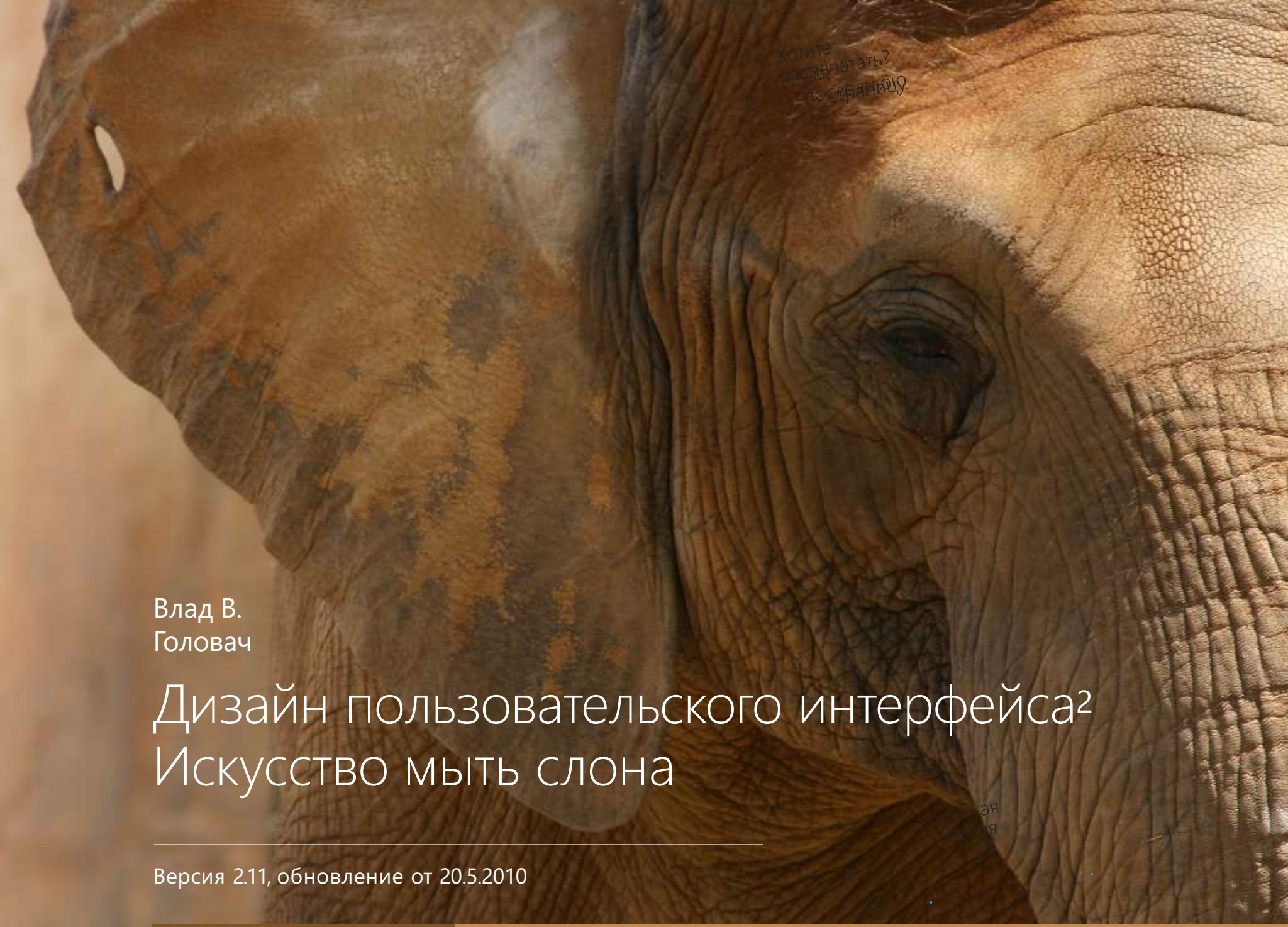


Хотите распечатать? Последнюю страницу.

Влад В.
Головач

Дизайн пользовательского интерфейса²
Искусство мыть слона

Свежая версия

Версия 2.11, обновление от 20.5.2010

Содержание

Предуведомление . . . 1

Введение: Искусство мыть слона . . . 5

Часть 1. Продуктивная работа . . . 8

Улучшайте, а не делайте хорошо . . . 9

Относитесь к своей работе пессимистично . . . 13

Стройте работу так, чтобы не думать о том, что именно вы могли забыть . . . 14

Переделывайте, а не думайте . . . 16

Начинайте работу с самой больной части . . . 19

Не ведите себя как эксперт . . . 21

Не философствуйте; общих ответов на общие вопросы всё равно нет . . . 23

Занимайтесь только той оптимизацией интерфейса, которая улучшает продукт . . . 25

Если вы профессионал, ведите себя профессионально . . . 32

Не занимайтесь тем, что у вас всё равно не получается . . . 37

Признайте, что вы перманентно и неизбежно ошибаетесь . . . 43

Часть 2. Что такое хороший интерфейс . . . 48

Так что же это такое?! . . . 50

Удобный, простой и «юзаб(и|е)льный» . . . 51

Эргономические показатели . . . 53

Дизайн, ориентированный на пользователей . . . 56

Дизайн, ориентированный на задачи пользователей . . . 59

Дизайн, ориентированный на мотивы пользователей . . . 61

Юзабилити . . . 64

Концепция деятельности . . . 70

Коммерческий успех . . . 72

Как использовать все эти знания? . . . 79

Заключение . . . 88

Рекомендуемая литература . . . 90

Об этой книге: как удобнее читать, напечатать, колофон . . . 94

Предуведомление

Первое предложение позаимствовано мной из замечательной книги Бертрана Рассела «История западной философии».

Чтобы эта книга могла избежать более суровой критики, чем та, которую она, несомненно, заслуживает, необходимо сказать несколько слов в качестве извинения и объяснения. Как извинение, так и объяснение нужны потому, что в этой книге вовсе нет того, что есть во множестве других книг по теме — а именно описания процесса разработки интерфейса и эвристик проектирования.

Здесь и далее под эвристиками я разумею эмпириче- ские правила, упрощающие поиск решений или огра- ничивающие набор возможных решений. Например, эвристикой является правило «в группе радиокнопок должно быть значение по умолчанию». Это правило не всегда верно, но проще и дешевле вовсе об этом не думать, а просто следовать этой эвристике.

- Дизайн интерфейса можно рассматривать как процесс. Авторы, пишущие о процессе, негласно допускают, что если мы провели этот процесс правильно (в правильном порядке выполнили нужные шаги, на каждом этапе сделали всё, что нужно), мы получаем правильный интерфейс. На мой взгляд, это допущение неверно. Поваренные книги тоже являются фактически сборниками процессов в кулинарии, но следование поваренной книге обеспечивает пристойный результат, только если её читатель умеет готовить вообще. Соответственно, одного процесса недостаточно.

Терминальная кнопка заканчивает взаимодействие пользователя с фрагментом интерфейса. Например, кнопки OK и Отмена в диалоговом окне закрывают это окно и поэтому называются терминальными.

- Другие книги исповедуют иной подход, а именно трактуют дизайн интерфейсов как следование набору правил-эвристик («терминальные кнопки должны быть размещены справа внизу или справа сверху» и т. п.). Авторы этих книг допускают, что если мы сделаем интерфейс по этим правилам, интерфейс получится хорошим. Этот вариант тоже не кажется мне плодотворным. Он напоминает мне изучение иностранного языка по словарю и грамматическому справочнику. Конечно, это

приемлемо, но истинного знания языка так получить невозможно. Например, фразы «Коля пошел в школу» и «В школу Коля пошел» являются правильными как грамматически, так и орфографически. Только вот смысл у них чуточку разный...

Отрицать важность этих двух подходов глупо. Но одних их, на мой взгляд, недостаточно. Дизайн интерфейса — творческая деятельность, которую невозможно алгоритмизировать хотя бы так же полно, как грамматику или поваренную книгу. Не менее важно обладать пониманием дизайна, умением отделять значимые вопросы в проектной деятельности от незначимых, а главное — отвечать на них.

Кроме того, дизайнер интерфейса редко когда занят разработкой хорошего интерфейса. На самом деле он *разрабатывает хороший интерфейс в имеющиеся у него сроки и бюджет*. Такая задача девальвирует и процесс (нужно знать ещё и КПД каждого этапа, чтобы решать, какие этапы проводить и сколько ресурсов на них выделить), и эвристики (без знания того, насколько применение каждой из них улучшает интерфейс, работать тяжело).

В этом смысле дизайн интерфейса не только и не столько процесс работы или знание и применение правил построения интерфейсов, а живая проектная деятельность. В ней не только важно сделать правильно, но не менее важно *не сделать неправильно* — и даже важно *не сделать чего-либо ненужного*. В ней нет чётких этапов (где это видано, чтобы жизнь следовала интеллектуальным построениям), но зато важен коэффициент полезного действия дизайнера и, к примеру, отсутствие трения в проектной команде.

Соответственно, в этой книге я пытался научить этой проектной деятельности. О процессе я не писал вовсе, а об эвристиках писал,

только если это эвристики поведения дизайнера, а не правила постро- ения интерфейса.

Не читайте эту книгу!

Неотъемлемое право любого автора — писать о том, что он считает важным, и делать это так, как он считает правильным. Обратной стороной этого права является то, что если довести его до максимума, получится текст, который будет неприятно читать другим.

Я это право до максимума довел.

Соответственно, прежде чем начать, я хочу предупредить о том, что ждет вас, тов. читатель, дальше:

- На момент написания этой книги уже существовал небольшой, но неплохой массив текстов по теме на русском языке (см. раздел Рекомендуемая литература). О том, что уже написано в этих книгах, я предпочел не писать вовсе.
- Существует довольно много людей, предпочитающих книги, в которых всё расписано до мелочей (желательно, чтобы последняя глава называлась «Полная Инструкция для Работы, в Которой Есть Всё и Которую Можно Прочесть и Дальше Совершенно Не Думать Головой»).
  Эта книга, мягко говоря, не такая.
- Я верю, что почти любой человек может спроектировать прекрасный интерфейс (всего-то нужно чуть-чуть развитого эстетического чувства, чуть-чуть эмпатии, чуть-чуть интеллектуальной честности), — но только если дать такому человеку *очень много времени*. Задача сделать прекрасный интерфейс в сжатые сроки гораздо более сложна и как таковая доступна не всем. Соответственно, я старался писать не о

Здесь и далее под эмпатией я понимаю не только и не столько способность эмоционально сопереживать другим, сколько умение представить себя на месте другого человека.

том, как сделать хороший интерфейс, а о том, как сделать интерфейс быстро и с максимальным КПД. Это сильно ограничивает изложение — о работах с низким КПД я просто старался не писать.

- Несколько раз я употребил обсценную лексику. Если вас коробит от невинного слова «какашка», немедленно переставайте читать — дальше будет только хуже.

Теперь вы знаете достаточно, чтобы решить, читать или не читать дальше.

Введение: Искусство мыть слона

«Писать книгу — это как мыть слона: трудно решить, с чего начать, и трудно понять, где слон уже вымыт, а где ещё нет».

Не знаю, кому принадлежит этот афоризм (поиск в интернете не дал результатов), но верность его я испытал на собственной шкуре. Это верно для книг и так же верно в отношении дизайна интерфейсов.

Все просто: слон большой, но и любой проект тоже большой. В нём много работы вообще и даже много разных видов работ. Нужно знать, *что* делать, *когда* делать, *когда переставать* это делать.

- Какую часть слона вы начнёте мыть первой?
- Когда сочтёте её чистой и перейдёте к следующей части?
- Как, собственно, убедить других, что слон уже вымыт (у них может быть и, по-видимому, действительно *будет* своё мнение по этому вопросу)?

Эти три вопроса порождают другие вопросы:

- Как определить чистоту слона?
- Какую степень чистоты можно считать достаточной?
- Каким КПД должна обладать ваша работа по мытью слона? И как этот КПД можно улучшить?

И ещё мириады других.

Так вот, на мой взгляд, процесс дизайна — это правильное задавание таких вопросов и поиск правильных ответов на них. Причем это не фигура речи — я давно убедился, что свободное творчество, не структурированное подобными вопросами, ужасающе непродуктивно. Чтобы ответить на эти вопросы, вам придется выбрать адекватный процесс и воспользоваться адекватными эвристиками. А поскольку в каждом конкретном проекте вопросов немного, трудно будет упустить что-то важное.

Хочу сразу отметить, что хотя слово «придется» выглядит угрожающе, ничего страшного в нем нет. В вопросах нет ничего исключительного. Если у вас есть понимание, что вы делаете и зачем, отвечать на них легко. А понимание, в свою очередь, получить несложно.

Я утверждаю, что если регулярно задавать самому себе по ходу проекта эти вопросы и максимально честно на них отвечать — вы сделаете хороший дизайн (в частности, интерфейс). Вопросы не дадут вам принять малоудачных проектных решений и заставят использовать правильный процесс.

Этим вопросам эта книга и посвящена. Она не может научить вас проектировать хорошие интерфейсы (и думаю, что ни одна книга этого не сможет), но зато научит оценивать их профессионально (а значит, и проектировать их), в частности:

- работать производительно, не совершая глупых ошибок
- ставить перед собой адекватные цели.

Эти навыки очень важны. Без них, к примеру, дизайнер может потратить большую часть времени проекта на улучшение незначимых свойств незначимых частей интерфейса, причем сделать это неприлично медленно; разругаться с заказчиком, но сохранять искреннюю веру в то, что всё было сделано правильно и что заказчик — придурок, ничего не понимающий в дизайне интерфейсов.

В начале моей карьеры всё это случалось и со мной, что было вредно как для нервов, так и для кошелька. Вам повезло — вы сможете научиться на чужом опыте.

Часть 1. Продуктивная работа

Сделать плохой дизайн можно не только плохим процессом или игнорированием (или неимением) знаний, но и неадекватной, а значит, непродуктивной деятельностью.

Я обнаружил, что существует несколько простых правил поведения (можете считать их заповедями), позволяющих избежать низкой продуктивности. Разумеется, список этих правил неполон — составить его может только существо безгрешное, а сам я, безусловно, не таков. Но лучше что-то, чем вовсе ничего.

Улучшайте, а не делайте хорошо

Большая часть артефактов, с которыми мы сталкиваемся, уже готовы. Архитектор *уже* спроектировал здание, в котором вы сейчас находитесь. Монитор, с которого вы сейчас читаете, *уже* сделан, как и стол, за которым вы сидите на *уже* сделанном стуле. Глядя вокруг, трудно не увериться, что все творения человеческих рук рано или поздно становятся готовыми.

Всё это оптический обман. На самом деле почти ничего не готово, а то, что таки готово, готово не потому, что оно действительно готово, а потому, что так получилось.

Подумайте о здании, в котором вы находитесь. Оно готово только в том смысле, что оно уже не станет другим, но с точки зрения построившего здание архитектора, оно, по видимости, далеко от идеала. Если архитектор научился чему-либо за время его проектирования, его собственная работа уже должна казаться ему несовершенной. Проблема архитектора в том, что после того, как здание построено, архитектор уже не может его улучшить. Разумеется, поумневший и ставший более опытным архитектор следующее здание построит несколько лучше, чем предыдущее, и так будет до тех пор, пока он не упрется в потолок своего профессионального роста. В этом смысле здание, которое вы считаете готовым, для построившего его архитектора — неизбежная итерация проектирования того самого, возможно ещё даже и не построенного здания, которое должно увенчать его архитекторскую карьеру. Невооруженного взгляда достаточно, чтобы понять, что зданий последней стадии мастерства меньше, чем зданий-итераций.

Это же соображение верно для монитора, стола и стула — разве что они ещё менее готовы, чем здание или блюдо. Монитор, стол и стул — товары массового шаблонного производства. Они производятся из года в год, поэтому их можно улучшать до бесконечности. Да, монитор уже готов — но только этот конкретный экземпляр и только для вас. Вполне возможно, что его производитель уже опознал несколько ошибок (или технологическая база улучшилась) — и уже продается новая, улучшенная модель. Причем здесь не идет речь о персональном опыте конкретного разработчика монитора — производитель может нанять другого инженера или дизайнера, и его навыки и мастерство позволят ещё немножко улучшить продукт.

Ничего готового нету; готовыми вещи становятся вынужденно.

В этом смысле производители не производят товары — вместо этого они создают и пестуют процесс производства, который, в свою очередь, продуцирует успешные товары, с каждым годом всё лучшие и лучшие.

См. в интернете о концепции Кайдзен (Kaizen). Например, https://en.wikipedia.org/wiki/Continual_improvement_process

Примеров тому множество. Корпорация Toyota внедрила т.н. экономное производство, при котором, в частности, безостановочно улучшаются все стадии технологического цикла — неважно, сильно или слабо, лишь бы было постоянное улучшение. Sony не ставит перед своими дизайнерами задачи сделать хороший продукт — вместо этого дизайнеры непрерывно улучшают продукцию предыдущего года.

Но это примеры сознательного применения этого подхода. Даже если его не осознавать, он действует всё равно, хотя не всегда эффективно.

См. https://www.bloomberg.com/globalbiz/content/jan2008/gb2008018_681920.htm

Например, каждая новая версия плееров iPod была лучше предыдущих, хотя, судя по впечатлениям владельцев, каждая версия была форменным совершенством. Лучше становятся вообще все плееры, хотя те же славящиеся низким качеством дизайна корейские производители улучшаются несколько медленнее. И это при том, что конечные потребители могут заметить только улучшения потребительских свойств продукта. Например, для потребителя всё равно, что Sony за время производства игровой приставки PlayStation 3 успела вдвое сократить её себестоимость (в основном за счет сокращения числа деталей), — хотя для Sony это, безусловно, значимое и *очень* приятное достижение.

Но, конечно, лучше этот подход всё-таки осознавать — т. е. признавать, что вам есть что улучшить и улучшить это надо — и использовать по максимуму. Если вы признаёте, что то, что вы делаете, никогда не будет готово до конца:

- Вам легче будет относиться к своей работе пессимистично (о пользе чего я буду говорить далее).
- Вы не будете тратить время на полировку того, что полировать ещё не нужно. Раз некоторые недостатки проекта можно обнаружить только после того, как он сделан до стадии неизменяемости (т. е. например, здание построено, а новая версия сайта выложена на рабочую площадку), полировать что-либо до посинения нужно только тогда, когда не осталось ничего менее готового. Например, непродуктивно полировать новую функцию программы, пока она ещё не выпущена и, соответственно, ещё не известно, как в действительности ей будут пользоваться. Проще и эффективнее просто реализовать эту функцию, выпустить продукт с ней, посмотреть на её реальную работу — и тогда уже эту функцию отполировать.

■ Наконец, отношение к продукту как к неготовому позволяет ещё и избежать вечной беды IT-индустрии — гонки за новыми функциями (creeping featurism). Субъективно готовый продукт незачем улучшать, поэтому он просто становится больше (за счет функций). А вот субъективно неготовый — улучшать можно и нужно, т. е. не только что-то хорошее добавлять, но и плохое исправлять.

День, когда вы только добавили что-то хорошее, но не исправили что-то плохое, – потерянный день.

Таким образом, при работе, прежде всего, нужно думать о том, как улучшить продукт, а не о том, чтобы сделать его хорошим. До конца хорошим он не станет никогда — но если вы будете относиться к нему как к неготовому, он станет достаточно хорош немного раньше, чем в противном случае.

Относитесь к своей работе пессимистично

Очевидно, что через_____, когда выйдет новая версия сайта
впишите период времени
или программы, над которой вы сейчас работаете, текущее состояние продукта будет казаться вам убогоньким — новая-то версия будет лучше. А раз новая версия будет лучше, текущая версия плоха.

Поэтому оптимизм в отношении своей работы явно вреден. Если вам нравится то, что вы сделали, вы явно обманываете себя, поскольку через некоторое время вам это разонравится (если же не разонравится, вы узнаете, что достигли вершины своего мастерства). Этот самообман, хоть он и очень приятен, опасен — если вы считаете что-то хорошим, у вас нет стимула пытаться сделать это ещё лучше.

Всё недостаточно хорошее – плохое. Поскольку улучшить можно всё, всё – плохо.

Поэтому пессимизм — верный и надежный двигатель профессионального роста, а значит и качества вашего дизайна. Признайте, что то, что вы сделали, вы сделали плохо, и у вас появится стимул сделать это лучше.

Самый простой способ достигнуть такого пессимизма — повесить над своим столом табличку со словами «Это можно сделать ещё лучше». Конечно, настроение она поначалу портит. Но качество работы — улучшает.

Стройте работу так, чтобы не думать о том, что именно вы могли забыть

Бóльшая часть работы дизайнера интерфейсов, во всяком случае, по затраченному времени — это думание. Чем сильнее мы ускорим это думание, тем быстрее и производительнее начнем работать. Эта задача может показаться невыполнимой (как же, для этого нужно стать умнее, что явно невозможно), но на самом деле она вполне реалистична — для этого всего лишь нужно изъять из нашего думания непроизводительные потери.

На мой взгляд, главной такой потерей является вопрос «О чём я забыл(а)?». Хороший интерфейс — явление многофакторное, об одном факторе вовремя не вспомнишь и всё придётся переделывать. Моё главное такое «достижение» было, когда я, нарисовавши прекрасный интерфейс главной страницы сайта, выверенный до мельчайших деталей, обнаружил, что забыл оставить в нём место для поля поискового запроса. Вуаля — целый день работы насмарку. Ошибка в пятисекундном рассуждении привела к совершенно неприличному объему потерянного времени.

Самый простой способ избавиться от таких потерь — всё записывать. Если перед тем, как рисовать интерфейс, вы напишете список всех его объектов — вероятность ошибки снижается на порядок. Если вместо того, чтобы долго разглядывать свеженарисованный интерфейс в поиске ошибок, просто прогнать его по контрольному списку — скорость проверки вырастет на порядок. Если записать перед началом проектирования интерфейса его желаемые эргономические качества

(такой перечень вряд ли будет больше абзаца), вы не забудете о них в процессе работы, а значит, все их и обеспечите (по мере сил).

Запись всего и вся — не единственный вариант решения. Если вы не умеете быстро писать, для вас он может оказаться неэффективным. Главное — построить свою работу так, чтобы вопрос «что забыто?» вообще не возникал. Например, через перманентные переделки.

Записывайте всё, в частности, совершенные вами ошибки. Нет лучшей гарантии, что вы их не повторите.

Переделывайте, а не думайте

Есть и другой способ ничего в интерфейсе не упустить. Можно даже и не пытаться всё учитывать — зато делать интерфейс быстро, пускай и плохим, а затем так же *быстро* переделывать его в хороший, а не долго думать/планировать/готовиться/изучать, стараясь сделать интерфейс хорошим с самого начала.

Результат в обоих случаях получится одинаковым. Различаться будет продолжительность работы — при хорошо поставленной технологии переделок и улучшений работа будет идти быстрее, поскольку вам не придется думать о том, что ещё вы могли упустить.

Кроме того, этот вариант предпочтительнее ещё и тем, что он более управляем. В случае долгого планирования сначала тратится много времени на думание, потом получается результат, но не факт, что достаточно хороший — и пока его ещё нет, вас (т. е. работника) нельзя перевести на другой проект, поскольку вы всё забудете. В случае перманентных переделок у вас быстро получается ужасный вариант и чуть позже приемлемый — и после этого работу над интерфейсом можно прервать или приостановить в любой момент. Работы такого вида очень любят руководители, поскольку она дает им большую гибкость в распределении сотрудников по проектам.

Наконец, работа через переделки лучше и тем, что её легче проверить. Когда вы долго ищете наилучшее интерфейсное решение, путь к нему содержится только у вас в сознании; он не воплотится в реальности, пока вы не доделаете работу до конца. Если же вы работаете через переделки, почти сразу вы получаете вариант вне своей головы (пускай только эскиз). Такой вариант можно показать коллеге или

заказчику, чтобы спросить у них совета. Две головы тут лучше одной, так что включение дополнительных людей в процесс дизайна улучшает качество.

Планируйте как можно меньше, начинайте проектировать интерфейс как можно раньше, зато переделывайте почаще.

Пара примеров такого рода работы из моей практики:

- Когда я разрабатываю интерфейсы сайтов, я полностью меняю вид навигации по сайту в среднем четыре раза за проект (всякий раз, когда обнаруживаются проблемы в текущей версии навигации).
- В одном из моих проектов общая структура интерфейса была настолько сложной, что я поначалу даже не пытался о ней думать. Примерно за месяц я нарисовал все фрагменты интерфейса и только в предпоследний день проекта, когда (наконец-то!) всё стало понятно, собрал все эти фрагменты в целостный интерфейс (разумеется, для этого их пришлось слегка доработать).

Чтобы начать так работать, вам потребуется хорошая технология прототипирования. Хорошая в том смысле, что она должна позволять побыстрее нарисовать первую, убогую версию интерфейса, но главное — обеспечивать максимально высокую скорость переделок.

На мой взгляд, сейчас лучшей такой технологией является сначала рисование прототипа на бумаге, а затем — финализация прототипа в Adobe InDesign. Изначально это верстальная система для полиграфии, так что использовать её для прототипирования

Подробнее о прототипировании интерфейсов в inDesign см. статьи на сайте моей компании.

интерфейсов — всё равно что забивать гвозди микроскопом. Но почему бы не забивать, если микроскоп ухватистый, а стоит не дороже молотка?

По крайней мере, сейчас InDesign лучше альтернатив — он легче в изучении, чем специализированные средства разработки, к тому же, хоть и уступая им в скорости создания первой версии, лидирует в скорости модификаций. Наконец, InDesign, в отличие от средств разработки вроде Adobe Dreamweaver или Microsoft Visual Studio, универсален — в нем можно запрототипировать любой графический интерфейс, будь то интерфейс программы или сайта.

Но какую именно технологию вы будете использовать — дело десятое. Главное — выбрать максимально скоростную и выжимать из этой скорости как можно больше переделок. Я вообще считаю, что пока не появилось, по крайней мере, третьей версии интерфейса, о каком-то его качестве говорить вообще нельзя.

Начинайте работу с самой больной части

Иногда существуют причины, когда важно оценить что-то полностью, в целом. Например, ресторанный критик должен оценить в ресторане всё: и обстановку, и обслуживание, и меню (и ещё много разных факторов). Но посетитель ресторана, решая, остаться в этом ресторане или вернуться ли туда ещё раз, оценивает ресторан гораздо проще — решение принимается по худшей составляющей, а не по среднему баллу. Соответственно, ресторатор, пытающийся улучшить своё заведение, должен, безусловно, улучшать в нём всё — но начинать ему стоит с самой худшей составляющей, просто потому, что её исправление принесёт максимальный эффект.

Это соображение кажется трюизмом. Я бы даже не стал писать об этом, но — до чего же трудно начать лечить то, что действительно болит, а не то, что первым попалось на глаза.

Перед тем, как приступать к работе, узнайте, что хуже всего сейчас и начните с решения именно этой проблемы.

У меня богатая практика в этом плане — полтора десятка дизайнеров, прошедших через нашу компанию, в своих первых проектах всегда начинали с того, что, вооружившись отвагой, бросались лечить то, что не болело вовсе или же болело совсем не сильно. Вероятно, в начале моей карьеры был таким же и я. Возможно, так же действуете и вы.

Поймите меня правильно — я не берусь утверждать, что гг. дизайнеры делали то, что делать было вовсе не нужно. Может, и нужно. Но первое впечатление заказчика, который приходит к дизайнеру именно потому, что у него что-то болит, было резко отрицательным. Вам бы тоже не понравилось, если бы вы пришли к хирургу вырезать болезненный нарыв, а он бы сначала отправил вас к окулисту выписывать очки (даже если бы вы действительно в них нуждались).

Соответственно, каждый проект надо начинать с определения того, что у интерфейса болит. Нужно как минимум расспрашивать заказчика, как максимум — тестировать, опрашивать пользователей и т. п. А затем формулировать список проблем, которые вы собираетесь решить своей работой. Если вы этого не сделаете, а сразу приступите к работе над интерфейсом, вы, несомненно, обидите заказчика, что непрофессионально.

Справедливости ради должен сказать, что иногда приходится начинать сразу с разработки, не выясняя сначала проблемы — если предметная область слишком сложна, лучше сразу бросаться делать интерфейс, поскольку при этом изучить задачу можно более предметно и быстро, а значит — дешевле для заказчика. Но опять-таки — при этом вы обидите заказчика, хотя и сэкономите его деньги.

Не ведите себя как эксперт

Всем известно, что «в грязи обитают мелкобы» и что они вредны для здоровья. Но это знание никак не коррелирует с количеством людей, моющих руки после посещения туалета (и тем более с количеством людей, моющих руки *перед* посещением). Знание есть у всех, а руки моет только малая часть людей.

Видно, что-то не так с этим знанием.

И я скажу, что именно не так — само по себе знание ~~приносит власть~~ бесполезно. Знание начинает приносить пользу, только когда оно превращается в конкретные действия.

Интеллектуальная ловушка здесь в том, что многие знания дают ощущение компетентности; кажется, что если я много знаю про дизайн интерфейсов, я могу расслабиться и всё равно получится хорошо.

Увы, не получится. В действительности многие знания ~~приносят печали~~ не сокращают объем необходимой работы, а, наоборот, его увеличивают. Например, раньше я проверял свежеразработанные мной интерфейсы на соответствие гораздо меньшему числу требований, чем проверяю сейчас — просто потому, что в начале моей карьеры я не считал эти параметры существенными или просто не знал о них.

Поэтому гораздо продуктивнее постоянно говорить себе, что «я ничего не знаю о дизайне интерфейсов». Эта установка ничего не сделает дурного с уже имеющимися у вас знаниями, но поможет избежать шапкозакидательства и в придачу откроет ваш разум для новых знаний (труднее учиться, если уже считаешь себя ученым).

Симптоматично, что даже я о нём написал, хотя, говоря откровенно, считаю это правило совершенно бесполезным для работы дизайнера интерфейсов.

И главное — надо делать, а не просто пассивно знать. Например, практически общеизвестно т.н. «правило 7±2», гласящее, что раз емкость кратковременной памяти человека редко бывает большей девяти элементов, делать меню большего размера неэффективно. Трудно прочесть хоть одну книгу о дизайне интерфейсов и не наткнуться на него. Но вот вы его узнали, и что же? Станут ваши интерфейсы теперь самопроизвольно лучше или нет? Нет, не станут. Чтобы это правило действительно помогало, вам понадобится включить правило «Ни в одном меню не более семи элементов» в свой контрольный список проверки интерфейсов и в дальнейшем не лениться проверять свои интерфейсы по этому контрольному списку. И без этой работы ваше абстрактное знание не стоит и гроша.

Если вы не превращаете свои знания в конкретные проектные шаги — это бесполезные знания.

Не философствуйте; общих ответов на общие вопросы всё равно нет

Мне регулярно задают вопросы класса «что лучше, интерфейсное решение А или интерфейсное решение Б?», например, «где лучше располагать корзину в интернет-магазине, сверху справа или в каком-либо другом месте страницы?». Каждый раз я страшно теряюсь, поскольку (печальный) опыт убедил меня, что:

- Универсальных решений, работающих всегда, почти нет. Существуют общие законы, следование которым в интерфейсе *всегда* продуктивно (например, законы Фиттса и Хика). К сожалению, открыта (и тем более доказана) только малая часть из них (собственно, только законы Фиттса и Хика), так что не будет преувеличением сказать, что мы, собственно, ничего про них и не знаем. Интересно при этом то, что все эти законы не говорят про конкретный интерфейс ничего — они описывают человека и особенности его восприятия/поведения, а не сами интерфейсы. Уверен, что так и будет продолжаться; что никогда человеческий разум не создаст, например, Общей Теории Корзины Покупок или Универсальной Парадигмы Чекбокса. А значит, не может быть и веры в возможность общих ответов на общие вопросы про интерфейс.

См. https://en.wikipedia.org/wiki/Fitts%27s_law и https://en.wikipedia.org/wiki/Hick%27s_law соответственно.

- Доказать работоспособность интерфейсного решения может только тестирование, а никак не интеллектуальные экзерсисы. Обилие факторов, влияющих на работоспособность интерфейсного решения, не позволяет ограничиться простым рассуждением при поиске общих ответов — рассуждение неизбежно будет длиннющим. А если рассуждение длительное, вероятность ошибиться возрастает на

порядки. Зачем искать ответы, заведомо зная, что они могут оказаться неверными? А вот если вы интересуетесь конкретным вопросом про конкретный интерфейс — всё как раз просто. Протестируйте его. Если тестирование покажет, что интерфейс работает, значит, он действительно работает.

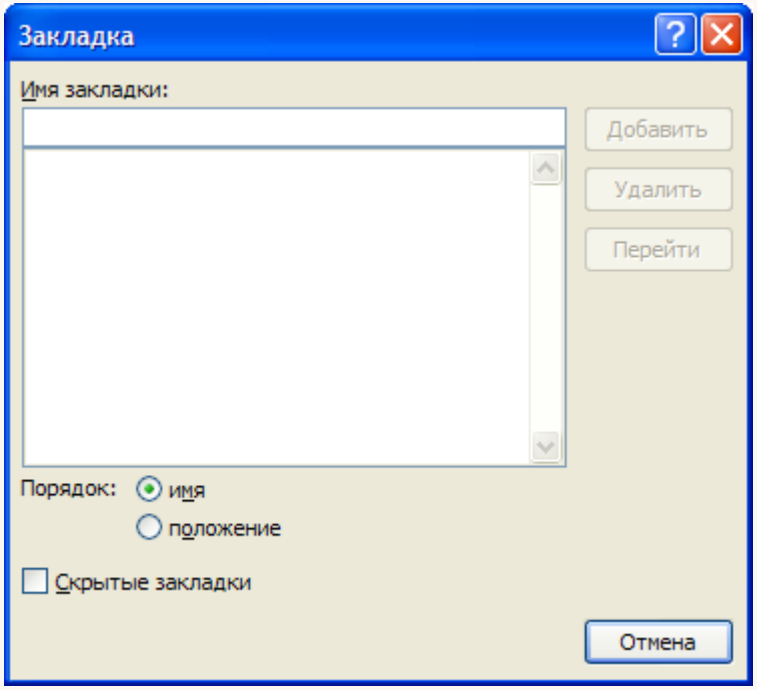
В этом смысле поиск ответов на общие вопросы просто бесперспективен — если вы уверены в верности ответа, вы, по всей видимости, ошибаетесь. Если не уверены, вы никогда не сможете найти верный ответ, потому что уже знаете слишком много «за» и «против» для каждого из возможных ответов.

Не задавайтесь общими, принципиальными проблемами, пока вы хотя бы трижды не нашли их частное решение.

Подводя итог, замечу, что, на мой взгляд, думание на общие, отвлеченные темы — занятие полезное, но не особо продуктивное. Гораздо продуктивнее задавать себе вопросы про конкретный интерфейс, а не про интерфейсы вообще.

Занимайтесь только той оптимизацией интерфейса, которая улучшает продукт

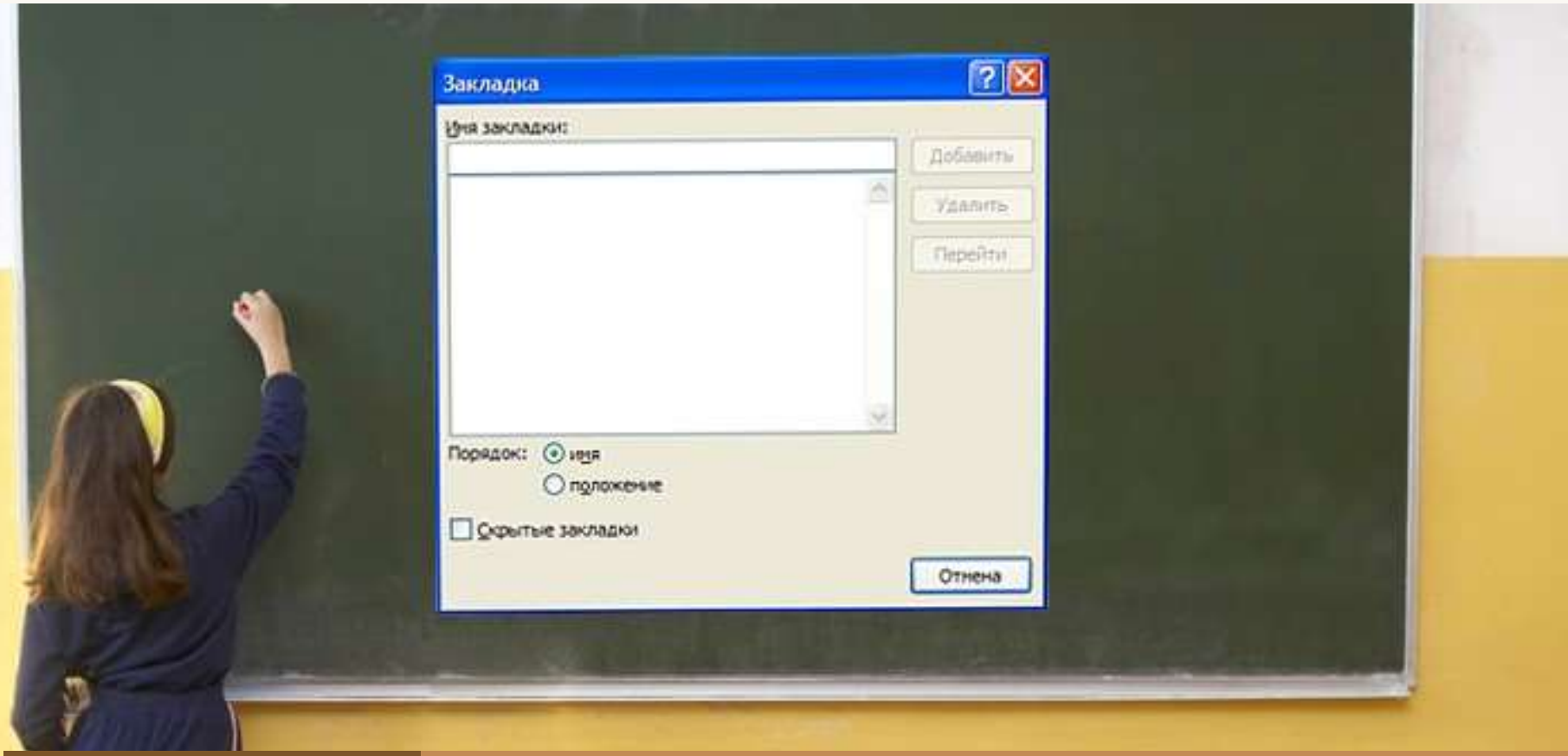
Вот диалоговое окно со списком закладок из Microsoft® Word 2007. Это окно не изменялось уже более десяти лет; точно таким же оно было в Word 97.



В этом окне есть неприятная интерфейсная проблема: оно очень маленькое, так что в него умещается всего несколько закладок и оно имеет фиксированный размер, который пользователь не способен изменить.

Для большинства пользователей эта проблема не важна, поскольку они закладками вообще не пользуются. Но для малой части аудитории эта проблема становится заметной: чем больше документ, тем удобнее работать с ним применяя закладки, и чем больше этих закладок сделать, тем будет удобнее работать. Но само окно содержит встроенный ограничитель этого удобства — если сделать много закладок, пользоваться ими будет неудобно, так как лишь несколько закладок умещаются внутрь окна.

Вот вам задание: найдите остальные интерфейсные проблемы этого окна (а пока вы ищете, не переходите на следующую страницу).



А теперь печальный факт — независимо от того, сколько проблем вы насчитали (лично я насчитал 4 мелких проблемы, не считая той, о которой шла речь на предыдущем слайде), *их число незначимо*. Просто потому, что все проблемы этого окна незначимы.

О важном и о неважном

Вот простая «формула» для оценки результата работы дизайнера интерфейсов:

Итоговое улучшение интерфейса = % пользователей затронутого фрагмента интерфейса × коэффициент важности этого типа аудитории × заметность улучшения качества интерфейса

Разберем эти составляющие:

«% пользователей затронутого фрагмента интерфейса» определяет частотность взаимодействия с этим куском интерфейса и количество пользователей соответствующей функциональности. Например, *все* пользователи сохраняют документы в Word, но только *некоторые* пользователи пишут в нём макросы.

«Коэффициент важности этого типа аудитории» — показатель того, насколько значим для успеха продукта тот тип аудитории, который взаимодействует с искомым фрагментом интерфейса. Например, можно постановить, что:

- обычные пользователи получают коэффициент 1
- спорадические пользователи, которые сами не покупают продукт (например, дети и любители пиратских копий), получают 0,5

- эксперты, влияющие на факт покупки продукта другими пользователями, получают 1,5
- начальство, подписывающее платежку магазину, получает коэффициент 5.

«Заметность улучшения качества интерфейса» — показатель улучшения интерфейса. Если пользователь, впервые увидевший новый интерфейс, начинает кричать от восторга, это одно. Если же пользователь просто думает про себя «Ну, давно пора было это сделать», это другое. А что, если пользователь просто не замечает улучшения? Предположим, что 1балл — это безразличная реакция пользователей на улучшение, а 5 баллов — просто-таки пароксизмы восторга.

Применим эту формулу для окна закладок. Предположим, что мы решили все его проблемы.

- Процент пользователей затронутого фрагмента интерфейса здесь совершенно мизерный, вероятно, не более процента пользователей вообще знают про это диалоговое окно. Ставим коэффициент 0,01.
- А вот коэффициент важности этого типа аудитории, наоборот, велик, поскольку это именно эксперты. Ставим 1,5.
- Заметность улучшения эргономических показателей интерфейса для невооруженного взгляда явно невелика. Никто не купит следующую, улучшенную версию, только потому что это окно стало лучше. Никто не будет даже особо рад, что оно стало лучше, хотя да, эксперты изменение заметят и одобрят. Можем постановить, что этот коэффициент улучшения стал 1,1.

$0{,}01 \times 1{,}5 \times 1{,}1 = 0{,}0165$

А теперь предположим, что мы *незначительно* (коэффициент 1.05) улучшили окошко открытия документа, которым пользуются *все* пользователи. Поскольку «все» здесь есть представители всех групп аудитории, результат будет *немного* другим:

$1 \times (1 + 0{,}5 + 1{,}5 + 5) \times 1{,}05 = 8{,}4$

Отчетливо видно, что маленькое изменение здесь дает в сотни раз больший эффект, чем большое изменение в малозначимом интерфейсе.

Получается, что итоговое улучшение продукта равняется 0,0165. Явно очень и очень мало. Разумеется, это число совершенно волюнтаристское, коэффициенты я назначал как бог на душу положит — но число говорит само за себя.

Разумеется, интерфейс стал лучше, так что работа по его улучшению не была полностью бесполезна. Но, разумеется, её никто не оценит.

Конечно, в идеальном мире (где всё сферическое и повсюду сплошной вакуум) это улучшение оправданно — много маленьких улучшений эквивалентны одному большому, так что всё равно что делать, если в результате станет лучше.

Но мир, в котором мы живем, не идеальный. В нём есть сроки/бюджет, начальство и маркетинг:

- Если сроки/бюджет сжаты (а они сжаты *всегда*), разумнее тратить время на работу, обладающую возможно большим КПД. Любой проект по оптимизации интерфейса сравнительно велик — он занимает недели работы и десятки|сотни тысяч рублей. Времени просто не хватает на работу, не дающую значительной отдачи. Думаю, никто не будет спорить, что в Microsoft Word был (и есть) резерв для улучшения, более значительный, чем окно закладок. Собственно говоря, разработчики Word уже доказали это, полностью переделав (и сильно, на мой взгляд, улучшив) систему меню в Word 2007.
- Любое начальство по своей природе будет запрещать работу, которая не дает заметного изменения — просто потому, что малозаметные изменения начальству труднее проконтролировать. Представьте, что разработчик пришел к своему менеджеру отчитываться о своей работе

и говорит «Я потратил полдня и улучшил окно закладок». Менеджер смотрит и не видит особых изменений. «Он меня дурачит!» — понимает менеджер и лишает разработчика премии.

- Маркетинг тоже склонен блокировать небольшие изменения, причем область его ковровой бомбардировки даже больше, чем у начальства — маркетинг одобряет только изменения, которые можно кратко объяснить словами. Понять маркетинг можно — ему придется продавать новую версию, а как это сделать, если сложно будет объяснить потребителям, в чём, собственно, заключаются улучшения. Потребители ленивы и нелюбопытны, так что список из трёх заметных и понятных всем улучшений будет гораздо приятнее маркетингу, чем список из трёх сотен улучшений мелких. Поэтому разработчику интерфейса, и так грустному от лишения премии, придется ещё и объясняться с человеком из маркетинга.

По этим трем причинам лично меня нисколько не удивляет, что у разработчиков из Microsoft за долгих десять лет не дошли руки отполировать это окно. Лично я их не виню.

Что делать? И чего не делать?

И здесь мы подошли к важной теме. Я убежден, что заметная часть неудачных проектов по дизайну или разработке пользовательского интерфейса объясняется прежде всего тем, что разработчики делают то, что делать, в принципе, можно, но — не сто́ит. В результате получают прорву организационных проблем, а в худшем случае — ещё и не успевают сделать то, что действительно важно для продукта.

Вот пара примеров из моей собственной практики (оба — трагичные):

- Однажды я все силы бросил на оптимизацию скорости работы пользователя, проигнорировав известный мне факт, что программой будут пользоваться полчаса в месяц и сокращения продолжительности сессии на 10 минут никто даже не заметит. Проект не оставил ни следа в графе *Прибыли*, хотя огненными буквами отпечатался в графе *Убытки*.
- В другой раз я сделал интерфейс очень лаконичным и простым, хотя догадывался, что лаконичный и простой интерфейс все будут считать признаком недостаточно мощного продукта. Клиент соглашался, что я в принципе прав и мой интерфейс объективно хорош, но выражал уверенность, что продать продукт с таким интерфейсом невозможно — никто нипочем не купит маломощный продукт. Проект не завершился.

Занимайтесь только той оптимизацией интерфейса, которая улучшает продукт.

Здесь я одумался и перестал делать то, что нравится, а во всё освободившееся время теперь стараюсь делать то, что нужно. Советую поступать так и вам. Начиная какую-то работу, следует спрашивать себя, улучшит ли это продажи, т. е. сделает ли эта работа лучше не только интерфейс, но и продукт.

Если вы профессионал, ведите себя профессионально

Представьте, что вы пришли к врачу с жалобами на страшные боли в боку. Проведя осмотр, врач говорит вам:

—Ну, у вас вот эта херовинка загноилась. Щас мы её шустренько оттяпаем, и будете вы как огурец.

Интуиция подсказывает мне, что вашим следующим действием будет немедленный побег от такого врача. Слишком уж непрофессионально он себя повел. Вот если бы врач сказал вам что-то вроде:

—У вас ярко выраженный случай синдрома Маркса-Спенсера, осложненный олигоптеризмом. Необходима немедленная операция. Если мы проведем её прямо сейчас, можете быть спокойны — в таких случах выздоровление практически гарантированно.

...вы бы покорно дали бы себя резать. Дизайнер интерфейсов похож на врача — к нему тоже приходят с жалобами, и его дело, по крайней мере, не оттолкнуть своим поведением клиента. Надо вести себя профессионально.

Легче, впрочем, сказать, чем сделать. Постоянно ловишь себя на том, что ведешь себя непрофессионально, например:

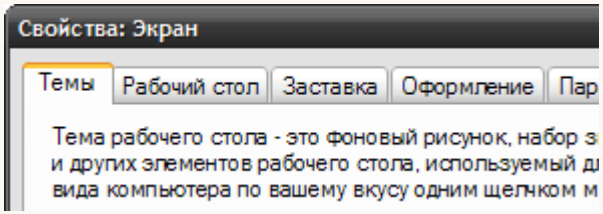
- используешь непрофессиональную лексику
- гарантируешь излечение
- потворствуешь пациенту.

Разберу эти пункты подробнее.

Непрофессиональная лексика

Что объединяет слова *выпадающий список*, *чекбокс*, *иконка*, *закладка*, *юзер*? Они непрофессиональны; если вы используете их, вы показываете своему заказчику, что вы не знаете даже профессиональной терминологии.

Выпадающий список не выпадает, а раскрывается, так что — раскрывающийся список. Чекбокс только в англоязычных странах checkbox, а у нас это флажок. Иконка это никакая не иконка, а значок (в интерфейсе и в пользовательской документации) или пиктограмма (в разговорах с заказчиком; слово уж больно звучное). Закладка бывает в книге, а то, что нарисовано в интерфейсе, называется вкладкой.



Это не закладки, а вовсе даже вкладки.

Во всех этих случаях единственно профессиональный метод наименования — пользоваться глоссарием. Глоссарий, в свою очередь, такая штука, что кто первый встал, того и тапки. Первой встала компания Microsoft — во всех их продуктах соответствующие понятия называются именно так, как написано выше (флажок, вкладка и т. п.), и что ещё важнее, называются именно так с незапамятных времен. Нравятся вам эти переводы или нет — но они привычны подавляющему большинству отечественных пользователей, а все другие наименования — нет.

Другое дело слово «юзер». Его использование — верный признак снисходительного отношения к пользователям (я, дескать, лучше этих мужланов, то бишь юзеров). На мой взгляд, при таком подходе сделать что-то хорошее этим пользователям несколько затруднительно.

Еще хуже дело со словом «юзверь». Скажу искренно — если вы его употребляете, вы *никогда* не сможете сделать хорошего интерфейса. С вами уже всё кончено; ваше сознание безнадежно ущербно.

Если на этом месте вы почувствовали горячее желание меня переубедить и доказать мне, что слово «юзверь» вполне приемлемо, воздержитесь.

Знание терминологии — навык приобретённый, а не врождённый. Соответственно, все дизайнеры поначалу говорят и пишут непрофессионально. Лечение, впрочем, просто, хоть и утомительно — надо

взять глоссарий Microsoft и заучить его наизусть (очень помогает работать только на локализованных версиях программ, если они есть).

Обещания, обещания

Известны случаи, когда после переделки интерфейс начинал работать лучше в разы и даже в десятки раз. Но это становилось известно только после переделки. Предсказать такое улучшение невозможно. Даже если предыдущую версию интерфейса делал совершеннейший придурок, всегда останется шанс, что *огромного* улучшения не будет.

Например, если речь идет об интерфейсе кассового аппарата, скорость работы пользователя определяется количеством действий в интерфейсе, а не самим интерфейсом. Не сократив число операций, не увеличить и скорость; а операции сокращать, чаще всего, некуда. В таких случаях почти всегда можно добиться ускорения в несколько секунд, но никак не на порядок.

В этом смысле лучше даже не говорить, что в вашей практике были громадные улучшения. Ну, были. Но если вы это говорите, вам придется объяснять почти всем следующим клиентам, почему в их случае вам не удалось добиться такого же улучшения (они-то выбрали вас именно потому, что вы такого улучшения когда-то достигли). Вдобавок вам придется объясняться ещё и со своими коллегами и конкурентами — придется очень убедительно доказывать, что улучшения добились именно вы, а не, например, дизайнер-график (он ведь тоже работал над интерфейсом новой версии). Если вы этого не сделаете, вам будут обеспечены диагноз «трепло» и презрение.

По этой самой причине так мало заслуживающих доверия историй об улучшениях интерфейсов. Понятную гордость своей работой дизайнеры и проектировщики держат при себе.

Поэтому ни в коем случае нельзя обещать клиенту значительных улучшений интерфейса. Процентов 10-15 ещё можно (этого можно достичь почти всегда), но не больше. Разумеется, нужно стараться добиться большего, но...

Лучше принести заказчику приятный сюрприз, чем неприятное открытие.

Если вы будете допытываться у врача гарантии своего выздоровления, врач постарается уйти от ответа, и даже если вы припрёте его к стенке, не даст вам стопроцентной уверенности. В конце концов, бывало, что люди умирали и от болезней, поначалу кажущихся обычным насморком.

Потворство пациенту

Если вы выписали больному таблетки, а он их не ест — это ваши проблемы. Больной-то уверен, что раз он уже побывал у врача, излечение ему обеспечено. И когда ему станет хуже, придет к вам ругаться (если не помрет; в этом случае придут родственники). Конечно, вашей вины в том, что пациент не принимает лекарства, нет никакой. Но проблемы будут и без вины.

Подобные случаи бывают и в практике дизайнеров интерфейсов. Иногда попадаются заказчики, которые понимают, что им нужен дизайнер, но не собирающиеся лечиться — т. е. готовые потратить деньги на оплату работы дизайнера, но имеющие собственное, и к тому же очень твёрдое, мнение о том, что именно и как надо сделать.

Необходимо явно и недвусмысленно объяснять таким заказчикам, что вам лично глубоко наплевать, будут они или не будут следовать вашим рекомендациям. Но если они не будут, вся ответственность ляжет на них самих и только на них, а вы, в свою очередь, ограничите свою роль получением оплаты за проект. Например, если вы обнаружили, что заказчик нанял дизайнера-графика, который портит ваш проект, вы должны немедленно отправиться к заказчику и честно сказать, что дизайнер халтурит по-черному и его нужно немедленно сменить или по крайней мере дать ему пару раз по голове. Да, вы испортите отношения с заказчиком — но вы испортите их в любом случае, когда выйдет продукт с убогим дизайном и за провал которого будут винить и вас. Однако если вы поссоритесь с клиентом сразу, будет, по крайней мере, шанс, что клиент выживет и впоследствии сможет понять вашу правоту. Если же вы будете ждать, шансов на это у вас не будет.

Конкретно такой случай был в моей собственной практике. По наивности я тогда попытался спустить вопрос на тормозах; в результате более месяца работы и прорва вдохновения отправились в помойку.

Не занимайтесь тем, что у вас всё равно не получается

Чтобы заниматься чем-то хорошо, нужен талант или, по крайней мере, предпосылки. Можно вполне профессионально рыть канавы, не имея особой мускулатуры, но эта работа не будет особенно эффективной (к счастью, от рытья канав мускулы нарастают очень быстро). Это, правда, сравнительно простая работа. Гораздо сложнее играть на рояле, если у вас нет слуха, с детства неправильно поставлены руки или же просто тремор такой, что вы и стакан-то держать не можете. Ещё хуже, если вы ленивы и нелюбопытны — тогда вы не сможете даже научиться играть *лучше*, не то что играть хорошо.

Это верно и для дизайна интерфейсов — если у вас нет таланта или хотя бы предпосылок, вы никогда не сможете делать это хорошо. Талант вы сможете обнаружить в себе со временем (или не обнаружить), о нем мне сказать особо нечего. Но о предпосылках сказать можно многое. Невозможно научиться делать роскошные интерфейсы, если вы:

Продолжение этого раздела я специально написал так жёстко и обидно, как только мог. Я убежден, что делать то, к чему нет способностей, трагедия и для самого работника, и для всех потребителей его работы. Лучше уж сразу всё понять и сменить область деятельности.

- не умеете говорить и, особенно, слушать;
- не имеете вкуса;
- не умеете писать;
- не умеете долго думать логично;
- нелюбопытны.

Неумение слушать и говорить

Мой опыт найма сотрудников свидетельствует, что большинство людей обладают очень неразвитой эмпатией, т. е. не умеют представить себя на месте другого человека. Без сопереживания просто невозможно узнать, что людям нужно (в частности, что нужно и чего не нужно пользователям ваших интерфейсов). Если вы не сможете это узнать, вы не сможете это людям дать. Просто, как дважды два.

См. http://en.wikipedia.org/wiki/Lake_Wobegon_Effect

Проблема с этим утверждением заключается в том, что, подобно тому, как все уверены, что у них есть чувство юмора, все уверены, что с эмпатией у них всё очень хорошо.

Проверьте себя. Если вы не согласны (полностью или частично) с утверждениями:

> Другие люди совсем не такие, как я. Их интересует не то, что интересует меня. Они мыслят иначе, чем я. Они знают то, что я не знаю. Они оценивают события по критериям, которые мне не известны и чужды.

И в любом случае, навыки эмпатии можно улучшить (это, кстати сказать, касается и меня самого).

...у вас, по всей видимости, проблемы с эмпатией. Придя на интервью с заказчиком, вы, скорее всего, слышите себя (а не заказчика), поскольку у вас нет стимула узнать что-то о другом человеке (я предполагаю, что для того, чтобы что-то узнать, нужно сначала признать, что это что-то ещё неизвестно). Это преодолимо, нужно всего лишь регулярно и методично напоминать себе, что другие люди — именно что *другие*. Кроме того, великолепно помогает тренировке эмпатии присутствие при юзабилити-тестировании своего интерфейса (можно смотреть записи его же; даже записи тестов чужих интерфейсов неплохо помогают тренировать эмпатию).

Отсутствие вкуса

Чтобы появилось само понятие красивого человека, большинство людей должно быть некрасивыми. Это же верно и для вкуса, с той только поправкой, что красота во многом есть свойство врождённое, а вкус — всегда приобретённый, никто не рождается с развитым чувством прекрасного. Его развитие занимает годы, и эти годы не просто надо прожить, но тратить время и усилия на развитие своего вкуса.

Вкус формируется постепенно. Лет двадцать назад мне случалось выходить замуж за мужчин, которых нынче я бы не пригласила к себе даже на обед.
Элизабет Тейлор

На мой взгляд, развитый вкус очень редок, хорошо ещё, если им обладают несколько процентов людей. Чисто статистически — вкуса у вас нет. А если это так, ваши интерфейсы (а они как-никак графические) будут уродливы, несмотря на все ваши усилия.

Проверьте себя. Если вы:

- в последние годы не были ни на одной художественной выставке
- не купили в прошлом хотя бы трёх альбомов по искусству
- не можете назвать хотя бы пять художников, которых вы особенно любите, и не можете объяснить, за что именно вы их любите
- не знаете названий хотя бы пяти архитектурных стилей и не можете объяснить, чем они отличаются друг от друга

...по-видимому, вы никогда не старались развить свой художественный вкус и, соответственно, у вас его нет. Идите в музей, прочтите несколько книг по искусствоведению, купите абонемент в консерваторию и т. п.

Неумение писать

Мысли имеют неприятное обыкновение из чётких и ясных становиться неоднозначными и неполными (а зачастую и самопротиворечивыми), стоит их только записать. Объясняется это просто — пока они ещё в голове, многие выводы, не говоря уж о противоречиях, попросту не видны. На бумаге же проблемы проступают немедленно.

Согласно Федору Тютчеву, мысль изречённая есть ложь. Мысль записанная тоже может быть ложью, но ложь эта, по крайней мере, не гарантированная, как в случае мысли высказанной.

На практике это обозначает следующее. Мы можем заметно повысить качество нашей мыслительной деятельности (а дизайн как раз является такой деятельностью), просто фиксируя свои рассуждения. Но это требует умения быстро писать, как технически (т. е. быстро набирать текст), так и творчески (т. е. оперативно составлять связный, читабельный текст).

Проверьте себя. В 2000 знаков опишите, что вы видите вокруг в этот самый момент, отмечая самое замечательное в том, что вы видите. Покажите текст кому-нибудь, кто, по вашему мнению, хорошо «знает русского языка». Если задание далось вам с большим трудом или знаток языка раскритиковал его в пух и прах, поздравляю — вы не писатель. Купите пару книг по редактуре (см. раздел Рекомендуемая литература) и практикуйтесь до посинения. Лично я научился хоть как-то писать только после семисот тысяч написанных знаков.

Проблемы с логикой

Чтобы были умные люди, большинство должно быть глупыми. Но ум, как и глупость, понятие растяжимое. Меня сейчас интересует только один сорт глупости, а именно неумение рассуждать о чем-то скучном *продолжительное* время.

Например, поиск в интерфейсе исключительных ситуаций (предпосылок для ошибок, как системных, так и человеческих) — совершенно необходимая стадия дизайна. Чисто технически — это очень продолжительное рассуждение, при котором дизайнер последовательно перебирает все объекты и состояния интерфейса в поиске возможных несоответствий («так, а что будет, если пользователь нажмёт кнопку поиска, не введя ничего в поле ввода?»). Многие люди (в том числе большинство т.н. гуманитариев) сталкиваются в этой работе с большими трудностями — нетренированному мозгу это очень тяжело и утомительно. Конечно, практика тренирует мозг очень быстро, но, поскольку эта работа такова, что её можно «задвинуть», отфутболив программисту, практики не получается. Отсюда — невозможность профессионального роста.

Проверьте себя. Найдите какую-нибудь книгу логических задач (особо рекомендую книги Раймонда Смаллиана) и попробуйте решать их, не отвлекаясь, не менее полутора часов подряд. Если дело не идет — найдите ещё десяток таких книг и решайте, решайте, решайте. Интеллектуальный марафон не повредит.

Отсутствие профессионального любопытства

Профессиональный рост на основе только собственной работы настолько долог, что, пожалуй, и невозможен. А чтобы интересоваться чем-либо за пределами собственной текучки, нужно любопытство.

Проверьте себя. Если вы:

- не знаете, кто такие Джозайа Веджвуд и Раймонд Лоуи
- не знаете, что такое интерлиньяж
- не можете перечислить хотя бы пять интерфейсных решений, которые в Windows лучше по сравнению с MacOS (и обратно)

...у вас скоро будут проблемы с полимерами (или уже). Решение очевидно: поставьте себе за цель каждый месяц читать хотя бы одну книгу по дизайну. Через год сами поймете, как развиваться дальше.

Что делать?

Подводя итог — все эти недостатки можно преодолеть (теоретически), но это потребует нешуточных усилий. Большинство людей страдают от этих недостатков, так что чисто статистически можно предположить, что они есть и у вас. Если вы не сможете их исправить, что ж, эта книга не пойдет вам на пользу — всё равно будете делать ерунду. Уж лучше заняться чем-нибудь другим.

Признайте, что вы перманентно и неизбежно ошибаетесь

Когда Сэмюэл Джонсон узнал, что один из его знакомых, несчастливый в браке и недавно овдовевший, женится снова, он охарактеризовал это явление как «триумф надежды над опытом». Такой триумф характерен для многих областей человеческой деятельности, но мне не известно ни одной профессии, в которой он был так же вопиющ, как в дизайне интерфейсов. Дело в том, что...

Не менее трети всех интерфейсных решений либо работают гораздо хуже ожидаемого, либо вообще неработоспособны.

Я выдвинул этот постулат на основе опыта тестирования как моих, так и чужих интерфейсов. Спешу заметить, что треть – это оптимистичная оценка.

Конкретно — независимо от того, насколько вы опытны и талантливы, как минимум треть всех разработанных вами интерфейсов (точнее, фрагментов интерфейса) работает гораздо хуже, чем вам бы хотелось. Отсюда — в работе дизайнера интерфейса неизбежно наступает момент, когда он уже вполне уверен в себе, но интерфейсы по-прежнему работают убого. Вуаля — встречаем очередной триумф надежды над опытом.

Удивляться тут нечему. Интерфейс — только половина во взаимодействии с системой, другая половина — человек, пользователь. Чтобы быть уверенным в том, что интерфейс работает хорошо, нужно с очень

высокой надежностью знать, что именно в любой конкретный момент пользователь:

- воспринимает в интерфейсе
- думает
- хочет добиться.

Очевидно, что это только мечта. Мы не можем даже с уверенностью утверждать, что понимаем наших близких, не то что совершенно чужих нам людей, тем более — многих людей, т. е. целую аудиторию.

Да, многое предсказуемо. Гештальт-психология, к примеру, узнала многое о восприятии. Но знаем мы по-прежнему очень мало. И непохоже, что в обозримое время узнаем достаточно, что, к примеру, появится чёткая и доступная практику Теория, Предсказывающая Любое Поведение Человека Или Группы Людей В Определенных Ситуациях.

Что делать?

В принципе, в том, что сделанный интерфейс работает не особо хорошо, нет ничего трагичного. Для дизайнера-практика неважно, насколько хорошо работает интерфейс, если переделанный интерфейс работает лучше, чем прежняя версия или же интерфейс, разработанный другим дизайнером.

Однако рано или поздно наступает момент, когда переделки интерфейса лечат часть эргономических проблем, но создают примерно столько же проблем новых (это отчетливо видно на любых зрелых продуктах с большим количеством версий; например, на программах

из Microsoft Office). В такой ситуации надо что-то делать. На мой взгляд, необходимо:

1. Признать, что интерфейс плох и будет плох, потому что плох я, дизайнер.
2. Начать интерфейсы тестировать.

Я делаю фигню

Постановка диагноза — первый шаг к излечению. Необходимо признать, что в наших интерфейсах всегда будут лакуны, поскольку мы не знаем, что думают и что видят в наших интерфейсах пользователи.

Повесьте перед своим рабочим местом табличку «В моих интерфейсах есть проблемы, о которых я ничего не знаю».

Это, знаю по опыту, довольно мучительная установка, требующая большой интеллектуальной честности. Как, например, отвечать заказчику, когда он спрашивает, что именно хорошо в свежеразработанном интерфейсе? Известно же, что в нём многое неработоспособно!

Тестировать, тестировать и ещё раз тестировать!

Отсутствие общей теории отнюдь не преграда для частного успеха. Например, нет Общей Теории Излечения Человеческого Организма, но врачи чаще лечат, чем убивают бездействием. Так же и у нас — работающий интерфейс можно получить даже без Общей Теории Интерфейсов. Надо сделать интерфейс так хорошо, как получается, после чего его протестировать, т. е. на практике узнать его слабые места и

исправить их. К Общей Теории мы так и не придем, но, по крайней мере, узнаем, где именно наш интерфейс отклоняется от идеала.

Без тестирования сделать действительно хороший интерфейс можно только случайно и только если интерфейс маленький.

Соответственно, про интерфейс только тогда можно сказать, что он хорош, если он протестирован и тестирование не выявило проблем.

Тестировать?

В принципе, можно и не тестировать. В самом деле, раз главным показателем качества интерфейса является качество продукта, а качество продукта, в свою очередь, определяется исключительно его продажами, можно просто подождать, пока продукт не выйдет на рынок, а тогда по продажам понять, хорош ли интерфейс или нет.

У этого подхода, впрочем, есть два недостатка:

- Если вы предложите заказчику так поступить, тот вас просто уволит — и правильно сделает, потому что при этом вы предлагаете заказчику рискнуть всем продуктом (т. е. большими деньгами и солидным временем), не предлагая никакой уверенности в успехе взамен.
- Из одного только объема продаж невозможно узнать, что именно плохо — слишком уж много факторов (цена, реклама, работа службы продаж, графический дизайн и т. д.) влияет на успех, выделить в (не) успехе сугубо роль интерфейса нипочем не получится.

Так что лучше уж тестировать.

Как тестировать

Я давно уже мечтал написать главу из одного единственного предложения, но не удержался вам об этом сообщить. Так что предложений тут три. Узнать почти всё о юзабилити-тестировании вы можете из моей статьи Юзабилити-тестирование по дешевке.

Часть 2. Что такое хороший интерфейс

Если дизайнер интерфейса не знает, каким должен быть интерфейс, ему не помогут ни волшебный процесс, ни магические эвристики — сделать хороший интерфейс он сможет только случайно.

Таким образом, вопрос «Что такое хороший пользовательский интерфейс?» является, безусловно, самым важным в работе. Тем более удивительно, что подавляющее большинство книг о дизайне интерфейсов вообще стыдливо молчат об этом. В лучшем случае упоминается юзабилити интерфейса (забегая вперед, замечу, что юзабилити никак не может быть единственным показателем качества). Понятно почему — универсального ответа на этот вопрос не существует, так что для каждого конкретного проекта нужен собственный ответ.

К стыду моему, в начале моей карьеры я и сам старался об этом не думать, полагая, что раз чёткого ответа нет, то и думать бессмысленно. Сейчас я считаю это грубейшей ошибкой, поскольку если в начале проекта не решить, к какому интерфейсу мы будем стремиться, неизбежно придется столкнуться со множеством проблем.

Например, будет невозможно:

- ...определить, какой именно аспект интерфейса наиболее неудачный. Следовательно, расставить приоритеты в работе также будет невозможно. Чтобы хорошо вымыть слона, очень полезно начинать с самой грязной его части, иначе к концу мытья грязь успеет снова распространиться по слону.
- ...понять, когда работа уже закончена и можно заняться чем-то другим. Слона *можно* мыть вечно, но зачем?
- ...объяснить другим, что с интерфейсом надо делать то-то и то-то, а не это и это. Например, практика показывает, что нельзя объяснить заказчику, что этот интерфейс уже хорош и без красной кнопочки, если перед этим не договориться с ним о том, чего мы хотим достичь. Если вы моете слона, начальник слонария *всегда* будет иметь основания утверждать, что слон грязный.

Так что же это такое?!

Увы, общего определения хорошего интерфейса, на мой взгляд, не существует. Однако для каждого конкретного продукта и проекта это определение вполне можно дать. Это определение будет состоять из набора качеств из разных концепций качества интерфейсов; отдельно списка желаемых качеств и отдельно списка качеств неважных или маловажных.

Пока просто перечислю эти концепции (дальше будет более подробно). Интерфейс хорош, если он:

- «удобный», «простой» и «юзаб(и)(е)льный»;
- обладает высокими эргономическими показателями;
- оптимизирован под своих пользователей;
- оптимизирован под задачи пользователей;
- оптимизирован под мотивы пользователей;
- обладает высокими показателями юзабилити;
- адекватен деятельности пользователей;
- коммерчески успешен.

При дизайне интерфейса важно добиться наилучшего решения согласно возможно большему числу этих концепций.

Удобный, простой и «юзаб(и|е)льный»

Если поискать в интернете, какими словами люди оценивают интерфейсы, окажется, что главными являются слова «удобный», «простой» и (только недавно вошедшее в обиход) «юзабельный» (или «юзабильный»). Эти слова очень интересны, поскольку они вполне адекватно отражают отношение человечества (точнее, его русскоязычной части) к интерфейсам. Безусловно, если интерфейс будет не «удобный», не «простой» и не «юзабильный», у него не будет никаких шансов.

Проблема в том, что никто не знает, что это такое, — уж больно растяжимые это понятия.

У слова «простой» есть ещё и та проблема, что простой интерфейс зачастую в принципе невозможен — у приборной панели пассажирского лайнера интерфейс будет посложнее, чем у мотодельтаплана, но это не делает его автоматически плохим.

По этой самой причине я настоятельно рекомендую вам просто забыть про эти слова, не думать ими и не использовать их в речи. Надо только

не забывать договариваться с заказчиком, что он тоже не будет их использовать.

> Практическое занятие: возьмите какой-нибудь интерфейс, лучше всего разработанный вами же, и попробуйте объяснить его достоинства какому-нибудь сослуживцу. Сослуживца же проинструктируйте в ответ на каждую вашу сентенцию отвечать «Ну это же неудобно!». Если вы продержитесь больше 5 минут, вы получите право с гордостью носить розовый берет.

Проще всего договориться с заказчиком/начальником, что он не будет использовать это слово, а вы взамен перед началом работы выспросите у него, какие именно интерфейсы он считает удобными и почему именно.

В любом случае, все три эти слова звучат крайне непрофессионально, так что пользоваться ими всё равно нельзя. Если вы употребляете их, вы оказываетесь в большой, но совершенно некомпетентной компании (чтобы убедиться в этом, просто поищите слово «интерфейс» в интернете).

В то же самое время вовсе игнорировать «удобство» и «простоту» нельзя — в конце концов, независимо от того, какой вы сделаете интерфейс, пользователи будут оценивать его именно в рамках этих понятий. Соответственно, если ваш интерфейс будет «неудобным» и «непростым», он явно будет плохим. Чтобы не допустить этого, интерфейс нужно оптимизировать под «простоту» и «удобство», но для этого нужно понимать, что конкретно люди вкладывают в эти понятия. Об этом будет идти речь далее, в главе про коммерческий успех.

Эргономические показатели

Оставив слова «удобный», «простой» и «юзабельный» для профанов, перейдем к более сложной концепции. Очевидно, что у любого интерфейса есть какие-либо эргономические показатели качества. Например, если интерфейс вынуждает своих пользователей совершать ошибки, этот интерфейс плох, во всяком случае, он заведомо хуже интерфейса, который помогает таких ошибок избежать. Потенциально можно собрать все эти показатели в одну систему.

Такого рода систем показателей довольно много, но наиболее распространенными являются т.н. показатели Шнейдермана, согласно которым интерфейсы характеризуются:

Подробнее см. Designing the User interface: Strategies for Effective Human-Computer interaction by Ben Shneiderman and Catherine Plaisant, Addison Wesley; 4 edition, 2004

- скоростью работы пользователя
- количеством человеческих ошибок
- субъективной удовлетворенностью
- скоростью обучения навыкам оперирования интерфейсом
- степенью сохраняемости этих навыков при неиспользовании продукта.

Эти показатели *очень* полезны:

- Они предметны и точны. Гораздо проще сделать интерфейс, с которым работать быстрее, чем интерфейс, который «удобнее» — мысли не разбегаются, критерии четче. Тем более просто объяснить другим, что интерфейс стал быстрее, чем «удобнее».
- Благодаря тому, что показателей качества несколько, растет производительность труда. Показатели помогают концентрироваться на отдельной задаче, не отвлекаясь и не фантазируя. Если потратить день на оптимизацию по скорости, потом ещё день на сокращение

операторских ошибок и ещё день на увеличение удовлетворенности, получившийся интерфейс будет лучше, чем интерфейс, на который потратили те же три дня, делая его «вообще лучше».

В то же время у показателей Шнейдермана есть и недостатки:

- Модель неполна, т. е. учитывает не все возможные показатели. Например, некоторые интерфейсы заметно способствуют усталости оператора, что во всех отношениях плохо. Конечно, усталость проявится в человеческих ошибках и в снижении скорости работы пользователя, но где она в модели?
- Показатели конфликтуют между собой. Например, скорость работы пользователя с определенного предела почти всегда начинает конфликтовать со скоростью обучения, т. е. попытки сделать *очень* быстрый интерфейс наталкиваются на ухудшение скорости обучения и обратно.
- Скорость обучения навыкам оперирования сложно оценить и тем более сложно измерить. В результате этот показатель существует только номинально и почти не используется в практической работе. Ещё хуже дела у сохраняемости навыков оперирования. Мало того, что этот показатель практически невозможно проверить, так и его актуальность для многих проектов вообще стремится к нулю. Например, зачем этот показатель интерфейсу почтового клиента, если с соответствующим интерфейсом и так работают каждый день? В результате из пяти показателей остаются четыре.
- На практике из четырех показателей Шнейдермана почему-то можно добиться высоких показателей только по любым двум, отбрасывая два оставшихся (например, вытянуть скорость работы и количество операторских ошибок, но потерять в скорости обучения и

удовлетворенности). Но из самой концепции непонятно, какие именно два показателя предпочесть в каждом конкретном случае. Например, можно прекрасно оптимизировать скорость работы пользователя, но потом окажется, что важнее была удовлетворенность (у меня однажды именно так и вышло).

- Модель ничего не говорит о том, какие показатели по каждой из шкал следует считать наилучшими из возможных в данном продукте.

Но достоинства показателей Шнейдермана всё-таки перевешивают недостатки. Простоту коммуникации с другими участниками проекта переоценить нельзя.

Дизайн, ориентированный на пользователей

Следующим подходом, как по сложности, так и по времени своего появления, является т.н. *дизайн, ориентированный на пользователей* (User Centered Design, далее ДОП). Его сущность кратко можно охарактеризовать следующим образом — если мы хорошо изучим нашу аудиторию и оптимизируем интерфейс под неё, наш интерфейс будет хорошим.

Отсюда два основных следствия ДОП:

- Отношение пользователей к интерфейсу является главным показателем качества интерфейса.
- Работа над интерфейсом невозможна без изучения особенностей аудитории, например, уровня начальной подготовки пользователей, их ожиданий, знаний предметной области и физиологических особенностей.

Например, молодые веб-дизайнеры (ещё) обладают хорошим зрением и поэтому не имеют ничего против очень мелкого набора. Адресаты их работы, как правило, несколько старше, и их зрение уже не так остро.

Отсюда проблемы.

Эти принципы действительно очень полезны:

- Интерфейс делается для пользователей, а не для его разработчиков. Поэтому мнение разработчиков, вообще говоря, неактуально для оценки качества интерфейса. Простейший вывод из этого соображения — если вы спроектировали интерфейс и он вам очень нравится, это ничего не решает, поскольку...
- ...пользователи могут отличаться и, как правило, действительно отличаются от разработчиков — и ДОП помогает об этом не забывать. Например, интерфейсы продуктов для детей разрабатывают взрослые,

при этом очевидно, что интерфейсы для детей разных возрастных групп должны отличаться друг от друга и тем более от интерфейсов для взрослых.

Если разработчик интерфейса будет оценивать его по собственным критериям, гарантированно получится плохой интерфейс. Другой пример: разработчик медицинского устройства может прекрасно разбираться в электронике и в интерфейсах, но маловероятно, что он так же прекрасно разбирается в медицине и даже вообще представляет особенности работы практикующего врача. Но предельные случаи как раз хороши именно тем, что они предельны и, соответственно, к ним можно подготовиться. В непредельных проектах, когда интерфейсы делаются для «обычных людей», т. е. таких же, как сам разработчик, почти всегда *неожиданно* оказывается, что аудитория всё-таки чем-то необычна.

Вот характерный случай из моей карьеры. Однажды мне нужно было разработать интерфейс для установки программы на сотовый телефон. Установка была сложной, поскольку, во-первых, включала настройки GPRS на телефоне для разных провайдеров, а во-вторых, могла выполняться на самых разных телефонах. Я потратил огромные душевные и интеллектуальные силы на эту работу, соответственно, она заняла прорву времени. В конце же оказалось, что целевые пользователи в нормальной ситуации вообще не устанавливают эту программу — они просят сделать это кого-нибудь более сведущего, например, своего шестнадцатилетнего племянника (для целевых пользователей конкретно этой программы было более естественно просить или заставлять, нежели делать самостоятельно). Для племянника же установка не представляла никаких проблем — после всех денег, которые он потратил на картинки и рингтоны. Если бы я знал это раньше, я бы ничего не стал бы делать вовсе и КПД моего безделья достиг бы заоблачных вершин.

Учитывая это, неудивительно, что ДОП ещё недавно был основным подходом к проектированию. О его распространенности можно судить хотя бы по тому факту, что аббревиатура UCD (User Centered Design) до сих пор часто употребляется для обозначения дизайна интерфейсов вообще.

Но у дизайна, ориентированного на пользователей, есть и недостатки:

- В системах, где пользователи получают деньги за работу с системой и эти пользователи легко заменимы, мнение пользователей важно лишь отчасти. Интерфейс должен быть производительным и эффективным, чтобы приносить деньги, а нравится он или не нравится — вопрос глубоко вторичный. Например, если идет речь о кассовом терминале,

Тут, правда, важно не увлечься. Сейчас кассиры, в принципе, расходный материал — их легко переобучить и у них довольно высокая текучка. Если экономическая ситуация изменится (на что я, кстати сказать, искренне надеюсь), тоталитарный интерфейс окажется, мягко говоря, неуместным.

важна скорость обслуживания, процент ошибок и невозможность воровства денег персоналом, а если кассирам терминал не нравится, это сугубо их проблемы, пускай увольняются, наймем других.

- ДОП не учитывает того простого факта, что человек — очень адаптивная система. Пользователи способны к адаптации в очень широких пределах. Например, я ни разу не видел кассира (а я интересовался вопросом), который бы ругал терминал, на котором он работает сейчас. Как правило, оценки варьируются от «намано» до «да хорошо всё». Но тот же кассир во время внедрения новой версии терминала будет верещать до посинения, потому что всё плохо. А через недели две — всё опять «намано». Как учитывать принципы ДОП в таких случаях? Ориентироваться на пользователя текущего, неадаптированного, или на уже адаптированного, но ещё не существующего?

Расходы и продолжительность исследования можно зафиксировать, но при этом не будет гарантии, что исследование даст результаты.

- ДОП требует исследований. Эти исследования, по сути, являются этнографическими. В свою очередь, этнографические исследования почему-то всегда дороги и длительны, вдобавок, почему-то всегда более длительны и дороги, чем запланировано (вероятно, потому, что этнографическое исследование применительно к интерфейсу — это всегда «найди то, не знаю что»). Поэтому включать в проектный цикл этнографическое исследование — верная гарантия того, что проект с управленческой точки зрения пойдет насмарку. Кроме того, до конца исследования совершенно непонятно, будет от него польза или нет, что депрессивно действует на всех участников проекта.

Суммируя, можно сказать, что ДОП негласно подразумевает: есть система (в частности — интерфейс) и есть её пользователи, которые и важны. Того, что делают пользователи в системе и ради чего они, собственно, с системой взаимодействуют, ДОП не покрывает. Зато его покрывает...

Дизайн, ориентированный на задачи пользователей

Когда имманентные проблемы ДОП стали несомненны, на смену старой концепции пришла новая. Согласно *дизайну, ориентированному на задачи пользователей* (Task Centered Design, далее ДОЗ), тот интерфейс хорош, в котором эффективно выполняются задачи пользователей.

Задача здесь — совокупность действий, в свою очередь являющихся совокупностями операций. Как правило, задачи могут быть решены несколькими разными способами, каждый из которых определяет свой набор действий. Например, девочка, которая хочет найти хорошего мальчика, может поставить себе задачу «прочесть Коэльо, чтобы не слыть дурой» или «купить новые стринги» (или обе эти задачи разом). Каждая из этих задач требует определенной, но достаточно свободной последовательности действий, так, она может купить книжку Коэльо в магазине, взять у подруги или прочесть в интернете. Дело дизайнера интерфейсов, согласно идеям ДОЗ — выбрать наиболее эффективное решение задачи и обеспечить её выполнение.

Этот метод несколько лучше ДОП, просто потому что шире и включает в себя ДОП (правда, в неявной форме): в самом деле, как мы выберем наилучшее решение для пользователей, если мы не знаем всего об этих пользователях? Но ДОЗ обладает и другими достоинствами:

- В отличие от ДОП, цели и методы которого не имеют прямого коммерческого выражения, дизайн, ориентированный на задачи пользователей, может быть оценён экономически. Например, если мы делаем интерфейс кассового терминала быстрее, мы можем посчитать,

насколько он стал выгоднее владельцу (нужно всего лишь посчитать полную себестоимость работы, которая высвободилась при улучшении).

- В отличие от потенциальных особенностей пользователей, число задач конечно и более предсказуемо при планировании. Благодаря этому проекты с ДОЗ значительно более управляемы, что очень приятно менеджменту.

Но у дизайна, ориентированного на задачи пользователей, есть и заметный недостаток. ДОЗ не позволяет определить, какое именно число решаемых продуктом задач является необходимым и достаточным. Всегда есть задачи, которые система (ещё) не решает; согласно ДОЗ, все их не просто можно, но даже нужно засунуть в интерфейс, поскольку чем больше задач будет решать система, тем она будет лучше. Отсюда бесконечный рост функциональности (т.н. creeping featurism).

Дизайн, ориентированный на мотивы пользователей

Преодолеть лавинообразный рост числа функций помогает другая концепция — *дизайн, ориентированный на мотивы пользователей* (Goal Centered Design, далее ДОМ; в отечественной практике его часто неправильно называют «дизайном, ориентированным на *цели* пользователей», переводя слово goal буквально). Согласно ДОМ, пользователи делают что-то для удовлетворения личных потребностей, иначе — мотивов; опознав эти потребности и сравнив их с задачами пользователей, мы получаем возможность лучше понять, что нужно делать.

Например, рассмотрим интерфейс программы-органайзера для секретарши. В настоящий момент она разговаривает по телефону с клиентом, который просит назначить встречу с её начальником на 10 утра пятницы. Что дают рассмотренные ранее концепции дизайнеру интерфейса для этой ситуации?

- Дизайн, ориентированный на задачи, в таких ситуациях ограничивается тем, что мы записываем в ТЗ, что «система должна поддерживать создание событий в календаре». Это очень хорошо; но для создания адекватного интерфейса этого явно недостаточно. Каким именно должен быть этот интерфейс? Непонятно.
- Показатели Шнейдермана тоже говорят немного. Очевидно, что этот интерфейс должен быть быстрым, продуцировать мало ошибок, быть простым в обучении и приятным. Но таким должен быть любой интерфейс!

Это – только пример, гипертрофированный для пущей наглядности; на самом деле я не мужская шовинистическая свинья.

- Дизайн, ориентированный на пользователей, может (а может и не, если во время исследования мы не обратили на это внимания) помочь, подсказав, что у секретарши такие длинные ногти, что она постоянно опечатывается при вводе с клавиатуры.
- А вот дизайн, ориентированный на мотивы пользователей, может сказать очень много. У секретарши основной мотив действий — не получить по голове. Очевидно, что если интерфейс будет не очень быстрый, так что клиенту на телефоне придется подождать лишних пять секунд, по голове секретарша не получит, а если получит — то не сильно. А вот если она вместо 10 часов назначит встречу на 11, один или несколько прицельных ударов ей обеспечены. Таким образом, ДОМ помог опознать важнейшую целевую характеристику нашего интерфейса — безошибочность. Соответственно, результирующее ТЗ на интерфейс должно звучать примерно как «интерфейс должен продуцировать минимум человеческих ошибок и быть по возможности быстрым» (скорость обучения и удовлетворенность здесь, к сожалению, придется выкинуть, поскольку из четырех характеристик интерфейса хороших результатов можно добиться только по двум).

Полезно составить список мотивов целевых пользователей и просто сравнить их со списком задач. Даже это помогает лучше оценить адекватность и полноту списка задач.

Этот пример показывает основную силу изучения и анализа мотивов пользователей: мотивы *объясняют* задачи. Благодаря ДОМ список

задач из ДОЗ, в лучшем случае отранжированный по частотности, приобретает собственно глубину и конкретику. Именно по этой причине в настоящее время дизайн, ориентированный на мотивы пользователей, является наиболее распространенным подходом к целеполаганию в дизайне интерфейсов (во всяком случае, обсуждается он чаще всего).

Юзабилити

К сожалению, все эти подходы (показатели Шнейдермана, ДОП, ДОЗ и ДОМ) довольно фрагментарны, поскольку концентрируются только на каком-то одном аспекте интерфейса. Общий подход, включающий в себя все эти концепции, очень желателен хотя бы потому, что всегда есть шанс о чём-то забыть или что-то пропустить. Такой целостный подход существует — это юзабилити.

Наиболее часто используемое определение юзабилити (из стандарта ISO 9241-11) гласит, что:

> Usability — the extent to which a product can be used by specified users to achieve specified goals with efficiency, effectiveness and satisfaction in a specified context of use.
>
> Юзабилити — степень эффективности, трудоемкости и удовлетворенности, с которыми продукт может быть использован определенными пользователями при определенном контексте использования для достижения определенных целей/мотивов.

В этой формулировке есть почти всё, что нужно для плодотворной работы дизайнера:

- ...эффективности, трудоемкости и удовлетворенности... — базовые показатели качества интерфейса (иные, чем у Шнейдермана, впрочем; см. далее).
- ...определенными пользователями... — внимание к аудитории, характерное для дизайна, ориентированного на пользователей.

- ...целей/мотивов... — целеполагание из дизайна, ориентированного на задачи, и дизайна, ориентированного на мотивы пользователей (ISO 9241-11 не особо разделяет задачи и мотивы, сваливая их в одну кучу).

  Т. е. в юзабилити-подходе есть все достоинства всех описанных мной ранее концепций, причем в придачу есть ещё кое-что, а именно среда (...при определенном контексте использования...).

  В самом деле, среда, в которой пользователи взаимодействуют с продуктом, может значительно влиять на интерфейсное решение. Например, если известно, что определенная операция всегда выполняется, когда пользователь разговаривает по телефону, есть все основания предполагать, что:

- Скорость взаимодействия очень важна, поскольку другой абонент не хочет ждать.
- Ещё более важна чувствительность продукта к операторским ошибкам, хотя бы потому, что работать и одновременно слушать, что тебе говорят — тяжело, особенно если говорят тихо. То же самое соображение применимо, если известно, что в помещении шумно.
- Если известно, что пользователь разговаривает по обычному телефону, без специальной гарнитуры, интерфейс надо проектировать только под ввод одной рукой (либо мышь, либо клавиатура), поскольку другой рукой пользователь будет держать трубку.

Таких средовых особенностей может быть много, например, почти в любом проекте важны:

- характеристики оборудования (например, скорость компьютера, разрешение, размер и цветопередача монитора, качество мыши, наличие или отсутствие достаточного места на столе, чтобы эффективно оперировать мышью)
- наличие или отсутствие карательной системы в работе пользователя
- шумовой фон и другие стрессогенные факторы
- внешние требования к скорости работы
- частота отвлечений.

Вот характерный случай из моей карьеры. Однажды я потратил довольно много времени на оптимизацию интерфейса под мышиный ввод, после чего узнал, что пользователи работают в столь тесных закутках, что формально наличествующей мышью им пользоваться очень тяжело (для мыши просто-напросто не было достаточно места).

Таким образом, концепция юзабилити более плодотворна для работы, чем перечисленные ранее концепции — в ней есть всё, что было там, и даже немножко больше. Вдобавок у неё есть и другое достоинство — это международный, вполне официальный стандарт (Россия тоже член ISO, так что скоро стандарт ISO 9241-11 станет и российским тоже).

Есть, увы, и проблема — выбранные ISO базовые характеристики (эффективность, трудоемкость и удовлетворенность), на мой взгляд, просто чудовищны:

- Существуют серьезные подозрения, что всего трёх показателей недостаточно, чтобы охарактеризовать все аспекты качества интерфейса. Шнейдерману, например, потребовалось пять штук.

Удивляться тут нечему. Глава группы создателей iSO 9241-11 не имеет заметного практического опыта разработки интерфейсов (в основном, ездит по конференциям; даже на своём собственном сайте он не приводит ни одного названия осчастливленного клиента). Кроме того, формулировки iSO во многом результат политических игр: заказчики хотят побольше и порастяжимей, поставщики — поменьше и почётче. В данном случае явно победили заказчики.

- Понять их невозможно, можно только запомнить. Я провел несколько дней, медитируя над их расшифровкой (в ISO 9241-11, помимо основного определения юзабилити, расшифровываются и все входящие в него понятия), и всё равно не могу сказать, что понимаю, что значит эффективность, а что — трудоемкость. Например, иногда мне кажется, что скорость работы пользователя это показатель эффективности, а иногда — что трудоемкости. У меня есть подозрение, что показатели специально выбраны такими, чтобы при желании в них можно было запихнуть всё что угодно (например, ту же степень утомления оператора при работе). Стандарт, который так легко растягивается, — гондон, а не стандарт.
- В любом случае, выбирать названия базовых показателей сходными до неразличимости (в оригинале efficiency и effectiveness) — страшный удар в спину переводчиков. Это *я* перевожу их на русский как «эффективность и трудоемкость»; распространен также перевод как «эффективность и продуктивность». Если кто-то переведёт efficiency и effectiveness как «эффективность и эффектность» — он *тоже* будет прав и этот перевод будет ничем не хуже остальных (в том смысле, что он будет так же труден для понимания).

iSO 9126 и iEEE 610.12-1990. Я не привожу их специально, чтобы не усложнять понятийный аппарат ещё больше.

- В другом стандарте ISO (и ещё в одном стандарте IEEE) есть иные определения юзабилити. Кому верить, спрашивается?
- В расшифровки понятий efficiency и effectiveness в ISO 9241-11 попали не только качества интерфейса, но и одна из особенностей продукта, а именно мощность — вероятно, стараясь интегрировать в понятие юзабилити проблематику дизайна, ориентированного на задачи пользователя, интегрировали и главную проблему ДОЗ, а именно ползучий рост функциональности. Уравняв по важности мощность продукта со скоростью работы пользователя, уравняли неуравнимое. Если

скорость работы пользователя это безусловное интерфейсное благо (трудно придумать продукт, для успеха которого пользователи должны работать медленно), то мощность продукта является достоинством относительным. Есть продукты, которые можно и нужно делать маломощными (например, чтобы сохранить их дешевыми или сберечь простой интерфейс). Получается, что ISO 9241-11 постулирует, что мощность повышает юзабилити. Сходу могу придумать как минимум один пример-опровержение — сверхмощный раскладной нож о двадцати лезвиях с отверткой обладает худшим интерфейсом, чем обычный кухонный нож.

Суммируя, я рекомендую вам забыть об efficiency и effectiveness как о страшном сне. Гораздо лучше заменить их соответствующими показателями Шнейдермана, чтобы получилось:

> Юзабилити — показатель количества ошибок, скорости взаимодействия с продуктом, скорости обучения навыкам взаимодействия и субъективной удовлетворенности определенных пользователей продукта, достигающих определенных целей/мотивов в определенном контексте использования.

Можно упростить ещё больше. Вместо загадочного «контекста использования» можно писать простое русское «среда»:

> Юзабилити — показатель количества операторских ошибок, скорости взаимодействия с продуктом, скорости обучения навыкам взаимодействия и субъективной удовлетворенности определенных пользователей продукта, достигающих определенных целей/мотивов в определенной среде.

Если вам кажется это слишком смелым (как же, моська лает на Международную Организацию по Стандартизации), я вас ободрю. Я такой не

См. Jakob Nielsen's Alertbox, Usability 101: introduction to Usability. http://www.useit.com/alertbox/20030825.html

единственный. Например, Якоб Нильсен, независимо разработавший точно такую же систему базовых показателей, что и Шнейдерман, вообще утверждает, что именно они и есть юзабилити. Почему бы ему не поверить? В конце концов даже одного взгляда на фотографию Нильсена достаточно, чтобы убедиться, что он знает о юзабилити кое-что и даже больше.

Концепция деятельности

Юзабилити — показатель количества операторских ошибок, скорости взаимодействия с продуктом, скорости обучения навыкам взаимодействия и субъективной удовлетворенности определенных пользователей продукта, достигающих определенных целей/мотивов в определенной среде.

Однако даже такая модифицированная формулировка юзабилити всё равно имеет определенные ограничения.

Во-первых, «Юзабилити — показатель...». Какое, собственно, значение этого показателя является необходимым и достаточным для каждого конкретного продукта? Когда уже можно прекращать мыть слона? Стандарт про это ничего не говорит.

Во-вторых, сведение юзабилити к единому показателю, с одной стороны, помогает работе, с другой — работе мешает. Мы можем измерить юзабилити, и это очень хорошо. Но как предсказать юзабилити во время дизайна? Как можно предсказать, насколько достигнутый показатель поможет хотя бы продажам?

В-третьих, все разобранные ранее концепции качества интерфейса очень уязвимы при недостаточном учёте факторов реальности. Например, если не учесть какой-либо мотив или задачу, интерфейс получится плохоньким (т. е. с низкой степенью юзабилити), это же верно и для средовых факторов, и для свойств целевых пользователей. Сами же эти концепции никак не помогают учитывать все значимые факторы, более того, в них неявно постулируется, что эти факторы уже известны.

Эти проблемы отнюдь не праздные. Например, Apple iPod, так, на минуточку, обладает низкой, по сравнению с конкурентами, мощностью и в придачу интерфейсом продуцирует огромное количество человеческих ошибок (из-за сверхчувствительного колеса управления). Но это ведь не мешает ему быть самым популярным плеером всех времен и народов! А его конкуренты, например, Creative, выпускающие мощные

плееры с зачастую очень эффективными интерфейсами, плетутся в хвосте. Видно, что что-то не так с определением юзабилити, раз конкуренты, моющие слона правильно, уступают Apple, моющей слона не по теории. В чём тут дело?

В последнее время концепцией, позволяющей решать такие проблемы, всё чаще считают концепцию деятельности, ведущую своё происхождение от работ отечественных психологов Выготского, Рубинштейна и Леонтьева. По многим причинам я намеренно не собираюсь касаться её здесь (слишком уж широкая штука, я бы сузил), однако порекомендовать её я считаю необходимым. Тем более что узнать о ней очень многое вы можете самостоятельно, просто поискав в интернете термины «теория деятельности» или, ещё лучше, «activity theory».

Коммерческий успех

Другой концепцией качества интерфейса является коммерческий успех. Хотя существуют интерфейсы для продуктов, в принципе не предназначенных для продажи, в большинстве случаев продукт надо продавать и его интерфейс должен помогать продажам. Если он это делает — это хороший интерфейс; если нет — плохой. Поскольку рынок наводнен успешными продуктами с дурными интерфейсами, приходится прийти к выводу, что есть какой-то изъян в концепции «для успеха продукта достаточно эргономичного интерфейса».

Этот изъян, по видимости, заключается в том, что эргономика рассматривает различные характеристики интерфейса как равно важные. Например, в концепции юзабилити показатели приводятся просто списком:

> Юзабилити — показатель количества операторских ошибок, скорости взаимодействия с продуктом, скорости обучения навыкам взаимодействия и субъективной удовлетворенности определенных пользователей продукта, достигающих определенных целей/мотивов в определенной среде.

Не может быть верной концепция, которая всегда дает один и тот же приоритет числу человеческих ошибок и удовлетворенности пользователей. Удовлетворенность, безусловно, коммерчески важнее. Без неё продукт попросту не купят и число ошибок вообще не будет важным. Незачем мыть дохлого слона.

Таким образом, хороший интерфейс — это интерфейс, который *нравится*. Интерфейс, который всем хорош, но, тем не менее, не нравится, интегрально плох.

Разумеется, это соображение не умаляет важности эргономики. Неэргономичный интерфейс снижает удовлетворенность; другой разговор, что зачастую это можно компенсировать иными способами. См. Три источника и две части субъективной удовлетворенности.

Например, т.н. дизайнерская мебель славится низким удобством (не говоря уж о низкой практичности). Но её покупают; правда, по причинам, слабо связанным с удобством.

Сексуальность

Термин «нравится» не очень продуктивен для работы дизайнера, потому что очень уж широк. По-моему, гораздо продуктивнее для оценки качества интерфейса слово «сексуальный».

> Хороший интерфейс — это сексуальный интерфейс. Сексуальный интерфейс — это интерфейс, который хочется иметь.

Сексуальность интерфейса трудно объяснить и аргументировать, но определить её наличие очень легко. Некоторые интерфейсы сексуальны, некоторые — нет. Даже не понимая, почему одни интерфейсы такие, а некоторые нет, мы легко можем понять, сексуален интерфейс или нет. Достаточно просто спросить себя: «Хочу я иметь этот интерфейс или нет?». Если нет, впереди ещё много работы.

Простота и удобство

Ранее я уже ругал слова «простой», «интуитивный» и «удобный» как критерии оценки интерфейса. Действительно, они слишком уж широки, растяжимы и субъективны, чтобы использовать их даже для коммуникации. Но с другой стороны — именно по этим двум характеристикам потребители первым делом будут оценивать продукт — и горе продукту, если он будет сочтен сложным и неудобным.

В действительности список оценочных терминов, т. е. конструктов, гораздо больше. Например, продукт может производить впечатление мощного или маломощного. Но эти конструкты варьируются от продукта к продукту; простота же и удобство используются в оценках всех без исключения продуктов.

Поэтому во время дизайна очень важно задавать себе два вопроса:

- Является ли получившийся интерфейс простым?
- Является ли получившийся интерфейс удобным?

Отвечать на эти вопросы, увы, несколько сложнее, чем при вопросе «Является ли получившийся интерфейс сексуальным?». «Простота» и «удобство» по терминологии Шалтая-Болтая являются словами-бумажниками, в них содержится больше смыслов, чем кажется на первый взгляд:

Аргументацию см. Bruce Tognazzini, intuitive vs. Familiar. http://www.asktog.com/columns/006intuitvsfamiliar.html

- Есть веские основания считать, что под словом «простой» применительно к интерфейсам надо понимать *уже знакомый* или *напоминающий что-то знакомое*. Субъективная простота или сложность имеет мало отношения к объективной сложности; в момент первоначальной оценки пользователь, ещё не знающий интерфейса, может только сравнивать его с другими интерфейсами, которые ему уже знакомы.

Во всяком случае — применительно к компьютерным интерфейсам.

- За словом «удобный», по моему глубокому убеждению, чаще всего скрывается «не требующий совершать большое количество непроизводительных действий», т. е. неудобный — это *трудоёмкий*. Похоже, что слово «трудоемкость» просто отсутствует в активном словарном запасе большинства людей, так что выражать своё мнение им приходится через слово «неудобный». Это мнение я составил на основе большого количества интервью с пользователями на протяжении ряда лет. В подавляющем большинстве случаев, когда потребители оценивали интерфейс как неудобный, глубокое интервью позволяло детектировать раздражение, вызванное большим количеством действий, не направленных напрямую на достижение результата (например, пользо-

ватели должны были снова и снова открывать один и тот же документ). В этом смысле очень важна проверка получившегося интерфейса на лишние или непродуктивные операции.

Проверяйте на оправданность все интерфейсные операции, удаляйте необязательные и исправляйте слишком уж трудоемкие.

Объясняемость хорошести интерфейса

Реклама, как известно, двигатель торговли. Насколько бы не был хорош продукт, его, по всей видимости, всё равно будут рекламировать, не дожидаясь того времени, когда потребители сами поймут его достоинства. В этом смысле интерфейс становится лучше, если делать его, в том числе, и в расчёте на будущую рекламу. Существуют как минимум два способа сделать продукт более «рекламировабильным»:

- *«Формулировабельность» улучшений.* Маркетингу требуются улучшения, которые можно легко объяснить несколькими словами. Поэтому не стоит заниматься работой по улучшению, которую вы сами не можете кратко охарактеризовать. Особая трудность, если делается новый продукт, который придется сравнивать не с прежней версией, а с конкурентами — некоторые вещи говорить о конкурентах неэтично (например, нельзя написать в рекламе, что «все остальные продукты — говно полное, а наш продукт — неполное»). Соответственно, чисто практически, перед началом всякой новой работы (например, закончив оптимизацию меню и переходя к оптимизации горячих

клавиш) надо просто тратить четверть часа на то, чтобы написать, чем интерфейс станет от этой работы лучше. Если описание такого улучшения получилось больше пятнадцати слов или есть подозрение, что оно будет понятно только другим разработчикам, работу лучше отложить, а вместо неё заняться чем-либо другим, что описывается лучше. Полезно также спросить маркетинг (если он есть), поможет ли им это улучшение. Если нет — грош цена такой работе. Буквально.

- *Картинка для прессы.* Это не так важно для сайтов, но очень важно для ПО: значительная часть продаж обусловлена публикациями в прессе, редакторы же постоянно мучаются поиском изобразительного материала. На практике, программа, у которой есть несколько *красивых скриншотов*, получает благодаря этому значительную фору — именно её скриншоты будут использованы в обзорах и других публикациях. К сожалению, привлекательность скриншотов имеет мало общего с красотой интерфейса — пользователи видят интерфейс в реальном размере и в динамике, а читатели прессы — в статике и в придачу в уменьшенном виде. Например, можно вбухать прорву времени в красивейшие тенюшечки и волшебнейшую анимацию — но ни то ни другое на скриншоте увидеть будет невозможно. Соответственно, работая над интерфейсом, нужно просто выделить некоторое время на то, чтобы сделать несколько экранов, выгодно смотрящихся на скриншотах.

И ничего страшного не будет, если пользователям эти экраны окажутся вовсе не нужны. Они не для пользователей, а для пишущих обзоры журналистов.

И да, я согласен, что это не особо этичная практика.

Хозяин — барин

Многие продукты продаются не своим будущим пользователям. Например, веб-интерфейс для почты выбирает администратор, а пользуются ей все. Можно сделать интерфейс для всех, но нет никакой гарантии, что он автоматически понравится администратору.

Непродуктивно и обратное — интерфейс, любезный сердцу администратора, вряд ли будет приятен пользователям. В таких ситуациях всегда есть (во многом этическая) дилемма — для кого делать продукт. Общего решения этой дилеммы нет, но надо постоянно помнить о том, что, потратив время на оптимизацию для всех, нужно выделить время и на оптимизацию для администратора — и обратно. Слишком уж часто я видел (часто на самом себе), как при планировании работы забывали о том, что интерфейс в таких случаях имеет две стороны, и в результате успевали улучшить только что-то одно.

Этика

Есть ещё одна причина считать, что коммерческий успех является самой важной характеристикой качества интерфейса. Работа на коммерческий успех, по-видимому, наиболее этична для дизайнера. Если продукт с разработанным дизайнером интерфейсом будет плохо (читай — недостаточно хорошо) продаваться по вине дизайнера, это проблема для всех. Для заказчика — потому что он заплатил дизайнеру денег в расчёте заработать на этой инвестиции. Для пользователей, поскольку если продукт будет неуспешен:

- Не появится шанса сделать продукт лучше в следующей версии. Или *ещё* лучше.
- Не появится шанса, что продукт улучшит чью-то жизнь. Например, неважно, насколько хорошо лечит вполне лечащее лекарство, если вы его не купили. У вас его попросту нет.
- Количество продуктов вообще сократится, т. е. сократится и выбор потребителей. Это может показаться очень сомнительным утверждением — но надо помнить, что большинство новых продуктов создаются предпринимателями, которые уже вывели в свет какой-то

продукт, а теперь инвестируют заработанные деньги и своё освободившееся время в новый бизнес. Чтобы это произошло, предыдущий продукт должен быть удачным, т. е. продаваться. Т. е. провалившийся продукт не позволяет появиться на свет другим продуктам.

Я много раз встречался с утверждением, что дизайнер интерфейсов это, дескать, «защитник пользователей», что дизайнер интерфейсов должен любить пользователей больше, чем себя, и защищать их от гнусных поползновений заказчиков. На мой взгляд, это очень сомнительная установка. Улучшать жизнь потребителей — это, прежде всего, дело самого предпринимателя. Именно за это он берет с потребителей деньги. Дизайнер же берет деньги не с потребителей, а с предпринимателя; соответственно, работать он должен именно что на предпринимателя. Кроме того, работающий по найму дизайнер рискует значительно меньшим, чем предприниматель, соответственно, он заведомо меньше вовлечен в процесс.

Кроме того, результаты установки на то, что «я делаю хороший интерфейс» несколько отличаются от результатов установки «я собираюсь осчастливить заказчика, разработав для него хороший интерфейс». В первом случае в проекте происходит слишком много трения. Из школьных уроков физики мы можем вспомнить, что трение замедляет движение и вызывает повышение температуры. Нужно оно вам?

Вы делаете довольным заказчика, для чего вам нужно сделать хороший интерфейс для пользователей.

Как использовать все эти знания?

Наконец, самое главное. Теперь вы знаете, что такое хороший интерфейс — осталось научиться применять это знание на практике. Это не так уж трудно. Всего-то нужно:

- Перед началом разработки в явной форме записать, какие эргономические характеристики важны для этого конкретного интерфейса. А в конце разработки проверить, выполнена ли поставленная задача; если нет — продолжать работу, если да — переходить к чему-то другому.
- Методически задавать себе заранее заготовленные вопросы в определенной последовательности.

Вопросы эти приходят из перечисленных мной ранее концепций качества интерфейсов. Например, из концепции показателей Шнейдермана приходят первые три:

1. Можно ли ускорить взаимодействие пользователя с этим интерфейсом?
2. Где в этом интерфейсе места, которые могут продуцировать человеческие ошибки? Можно ли изменить эти фрагменты?
3. Что в этом интерфейсе не способствует обучению? Что пользователю нужно знать, чтобы успешно взаимодействовать с этим интерфейсом? Есть ли в этом перечне что-то, чего сам интерфейс не сообщает пользователю?

Эти три вопроса нужно задавать себе по очереди. Если после ответов видно, что интерфейс надо менять, остальные вопросы нужно задать себе снова после переделки. Если на все три вопроса удалось дать

отрицательный ответ, переходим к следующей порции вопросов из остальных концепций качества:

4 Известно ли мне о пользователях что-нибудь, что делает этот интерфейс плохим?

5 Удовлетворяет ли этот интерфейс все известные мне мотивы пользователей?

6 Совместим ли этот интерфейс со средой, в которой работают пользователи?

7 Если и по этим вопросам всё хорошо, переходим к проверке, как выполняются в интерфейсе задачи пользователей. Соответственно, этот вопрос звучит как «Есть ли задачи, которые неэффективно отрабатываются интерфейсом?». Как правило, достаточно проговорить вслух (а ещё лучше написать), как в этом интерфейсе пользователь выполняет все свои задачи (лучше всего писать о себе, а не о абстрактном пользователе, например «Из меню Документ я открываю окно настроек зета-преобразования, ввожу значение 40 в поле Количество человеков, затем открываю...»). Как правило, такая проверка выявляет множество несоответствий или попросту пропущенных кусков.

Если это произошло, возвращаемся к самому первому вопросу. Если нет, задаем себе последний вопрос:

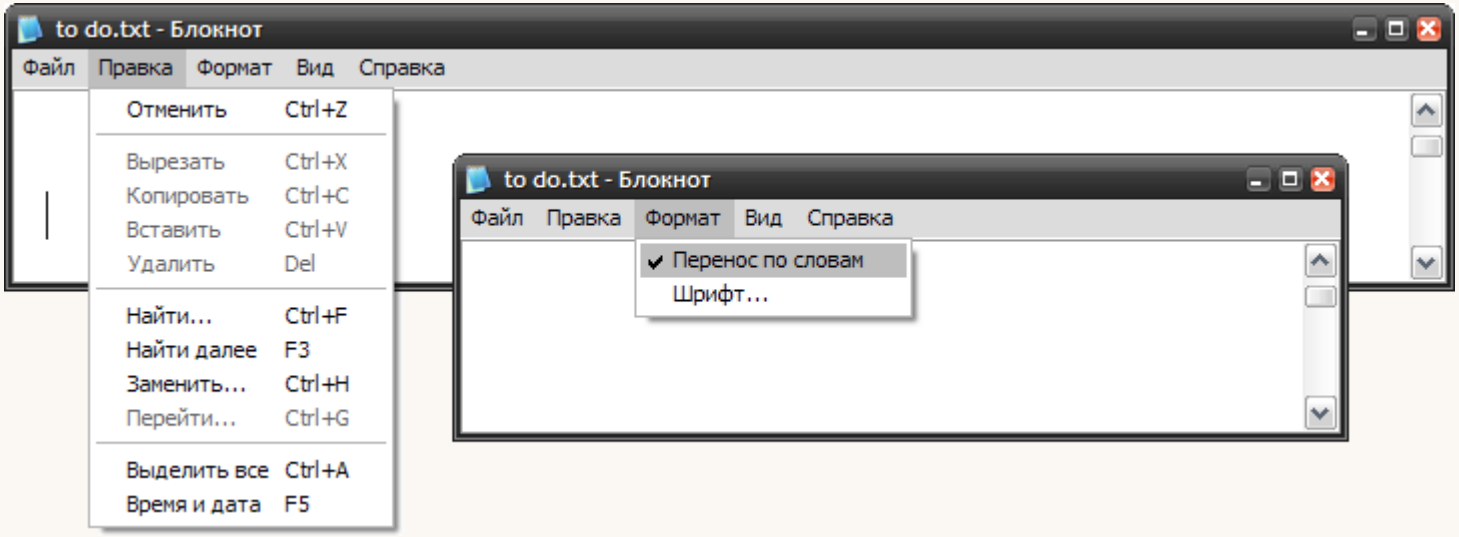
8 Сексуален ли этот интерфейс и можно ли его сделать ещё сексуальнее?

Очевидно, что вопросов можно сформулировать и больше. Я и сам пробовал пользоваться бóльшим набором вопросов, но мне не понравилось.

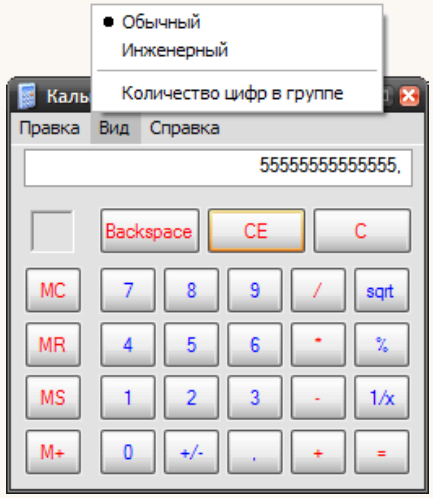
Как видите, вопросов всего восемь и в них нет ничего особо страшного. Есть только одна хитрость: у любого продукта много функций и, соответственно, цельных «кусков» интерфейса.

Например, у обычного Блокнота из Windows — на что уж малюсенькая программулька — пять уникальных функций, не считая стандартной функциональности программ Windows:

- функция — вставка времени и даты
- функция — переход на строку по её номеру
- настройка — переносить ли слова на новую строку
- настройка — показывать или не показывать строку статуса окна
- настройки — как показывать текст (выбор шрифта, кегля и т. п.).



Все эти функции — фрагменты интерфейса, для каждого из которых нужно задавать себе эти вопросы отдельно. Только после того, как вы ответите на все вопросы про отдельные фрагменты, можно задавать себе их о программе в целом. Без этого ваши ответы не будут особенно глубоки.



Оставим на совести локализаторов качество названия элемента.

Проиллюстрирую эту методику на примере другой программы из поставки Windows — Калькулятор (только для обычного режима, разбор инженерного режима только удлинит изложение). Предположим, заказчик принес мне этот интерфейс и я должен его улучшить.

> Задание для самопроверки: самостоятельно составьте список всех проблем этого интерфейса, во всяком случае, список всего, что бы вы изменили. Затем сравните свой список с результатами моего анализа.

Сначала разделим этот интерфейс на фрагменты для отдельной проверки. Ими являются основное меню, показ результатов и сама панель с цифрами. Соответственно, мне нужно задать себе 32 вопроса: 24 для отдельных фрагментов интерфейса и ещё 8 для программы в целом.

Начнем с меню. Его единственным нестандартным элементом является переключатель «Количество цифр в группе». Если его включить, длинные числа будут делиться на части по три цифры. Начинаем задавать вопросы:

1. Можно ли ускорить взаимодействие пользователя с этим меню? — Нет.
2. Где в этом меню места, которые могут продуцировать человеческие ошибки? Можно ли изменить эти фрагменты? — Название пункта «Количество цифр в группе» затруднительно сделать совершенно понятным. Можно, конечно, переименовать его в «Разделять длинные числа на группы», но это очень длинно. Может быть, пункт стоит выкинуть из меню, включив деление по умолчанию?
3. Что в этом меню не способствует обучению? — Если выкинем элемент «Количество цифр в группе» — ничего.

4 Известно ли мне что-нибудь о пользователях, что делает это меню плохим? — Нет.

5 Удовлетворяет ли это меню все известные мне мотивы пользователей? — Да.

6 Совместимо ли это меню со средой, в которой работают пользователи? — Да.

7 Проговариваем список всех задач, которые пользователь может решать с помощью меню. Вроде бы ничего проблематичного нет.

8 Сексуально ли это меню? — Нет, не сексуально. Стандартное вообще не может быть сексуальным. Но здесь это и не нужно.

Перейдем к показу вывода результата:

1 Можно ли ускорить взаимодействие пользователя с полем вывода? — Очевидно да, поскольку длинные числа медленно сканируются взглядом. Нужно включить режим разбиения длинных чисел по умолчанию.

2 Где в этом поле места, которые могут продуцировать человеческие ошибки? Можно ли изменить эти фрагменты? — Если пользователю нужно прочесть результат вычислений, а не просто скопировать его в другую программу, показ длинных чисел сплошняком может вызвать ошибки. Нужно включить режим разбиение длинных чисел по умолчанию. Кроме того, полезно увеличить размер цифр, чтобы улучшить их разборчивость. Наконец, ошибки в продукте такого типа чаще всего обнаруживаются слишком поздно. Текущий интерфейс не помогает проверить результаты своих вычислений: единственный способ самопроверки — повторить расчеты и сравнить результаты, что неоправ-

данно долго. Нужен какой-либо механизм самопроверки, например, можно показывать промежуточные результаты вычислений.

3 Что в этом поле вывода не способствует обучению? — Вроде ничего.

4 Известно ли мне о пользователях что-нибудь, что делает это поле плохим? — Нет.

5 Удовлетворяет ли это поле вывода все известные мне мотивы пользователей? — Да.

6 Совместимо ли это поле со средой, в которой работают пользователи? — На мониторах с большим количеством точек на дюйм (например, на многих современных ноутбуках) цифры могут быть настолько мелкими, что будут трудночитаемы. Стоит увеличить.

7 Проговариваем список всех задач, которые пользователь может решать с помощью блока показа результата. Вроде бы ничего проблематичного нет.

8 Сексуален ли этот интерфейс? — Нет, не сексуален, поскольку стандартен, но это ничего не стоит изменить: например, увеличить кегль у цифр или выбрать шрифт со специфическими цифрами. Или сделать и то и другое.

Закончим анализом панели с цифрами:

1 Можно ли ускорить взаимодействие пользователя с этой панелью? — Маловероятно.

2 Где в этой панели места, которые могут продуцировать человеческие ошибки? Можно ли изменить эти фрагменты? — Разборчивость кнопок умножения и вычитания (пиктограммы * и –) не очень высока, что может продуцировать ошибки. Увеличить размер пиктограмм в кнопках арифметических операций.

3 Что в этой панели не способствует обучению? — Названия кнопок MC, MR, MS и M+ ничего не говорят пользователю, если он не знает их назначения. Это нормально для инженерной версии калькулятора, но неприемлемо для обычной. Стоит увеличить размер кнопок, чтобы в них влезли лучшие названия (или вообще отказаться от них, поскольку всё равно есть буфер обмена). То же, хоть и в меньшей степени, касается кнопки sqrt. Либо увеличить, либо снабдить пиктограммой квадратного корня. И опять — чем отличается кнопка C от кнопки CE? Может быть, эту CE можно внедрить в поле вывода результата?

4 Известно ли мне о пользователях что-нибудь, что делает этот интерфейс плохим? — Пользователи явно пользуются этим интерфейсом крайне спорадически (сложные вычисления всё равно придется делать в инженерной версии калькулятора, а для частого счета удобнее настоящий калькулятор с крупными клавишами, дающими тактильную обратную связь). Непотребные термины на кнопках из предыдущего пункта явно не подходят для вечно малоопытных пользователей.

5 Удовлетворяет ли этот интерфейс все известные мне мотивы пользователей? — Да.

6 Совместима ли эта панель со средой, в которой работают пользователи? — Нет; как минимум для новых мониторов с высоким разрешением и небольшим размером экрана он не подходит — слишком мелкие элементы управления (их размер оптимизировался во времена 15-дюймовых экранов на 800x600 пикселей).

7 Проговариваем список всех задач, которые пользователь может решать с помощью панели клавиш. Вроде бы ничего проблематичного нет.

Задавая себе эти вопросы, всегда нужно помнить, что есть риск дать неверный, или, чаще, неполный ответ. Например, разбирая интерфейс калькулятора, лично я упустил такую распространенную задачу, как расчет с использованием скобок. Традиционные калькуляторы не умеют его выполнять (скорее всего, традиционный дизайн сложился, когда регистры памяти, нужные для скобок, были непозволительной роскошью). Кнопки MC, MR, MS и M+ (и индикатор использования памяти) были неуклюжей попыткой решить эту (в частности) проблему. Для нашего калькулятора такой расчет не проблема, так что по-хорошему, кнопки MC и MR надо просто заменить на кнопки со скобками — но я этого не заметил. Защититься от этой проблемы можно, только если над интерфейсом работают несколько человек.

8 Сексуальна ли эта панель? — Нет, не сексуальна. Впрочем, непонятно, как это можно исправить.

Наконец, пришло время задать вопросы относительно всего интерфейса в целом. У нас уже получился довольно большой список правок, так что первые восемь вопросов для экономии времени можно пропустить.

> Задание для самопроверки: ответьте на эти вопросы относительно калькулятора в целом и сравните свои находки с моим итоговым списком изменений.

Итак, в программе Калькулятор стоит, как минимум:

1 Показывать результаты вычислений разбитыми на группы цифр (317543 => 317 543) по умолчанию, убрав соответствующий элемент меню.

2 Увеличить размер цифр в поле результатов.

3 Увеличить разборчивость кнопок математических операций.

4 Прибить кнопки операций с памятью, но зато вставить кнопки для скобок и что-то сделать с кнопкой квадратного корня.

5 В идеале — при запуске спрашивать у ОС разрешение экрана и увеличивать размер всех элементов, если разрешение слишком велико.

6 Реализовать показ промежуточных результатов калькуляции.

7 Сделать окно всегда плавающим поверх других окон (настройкой). У этого интерфейса есть проблема: если нужно сделать серию расчетов, копируя результаты в другое окно, окно калькулятора всё время будет пропадать, перекрываясь окном, в которое копируются результаты. Пользователю придется всякий раз тратить время на возвращение в окно калькулятора.

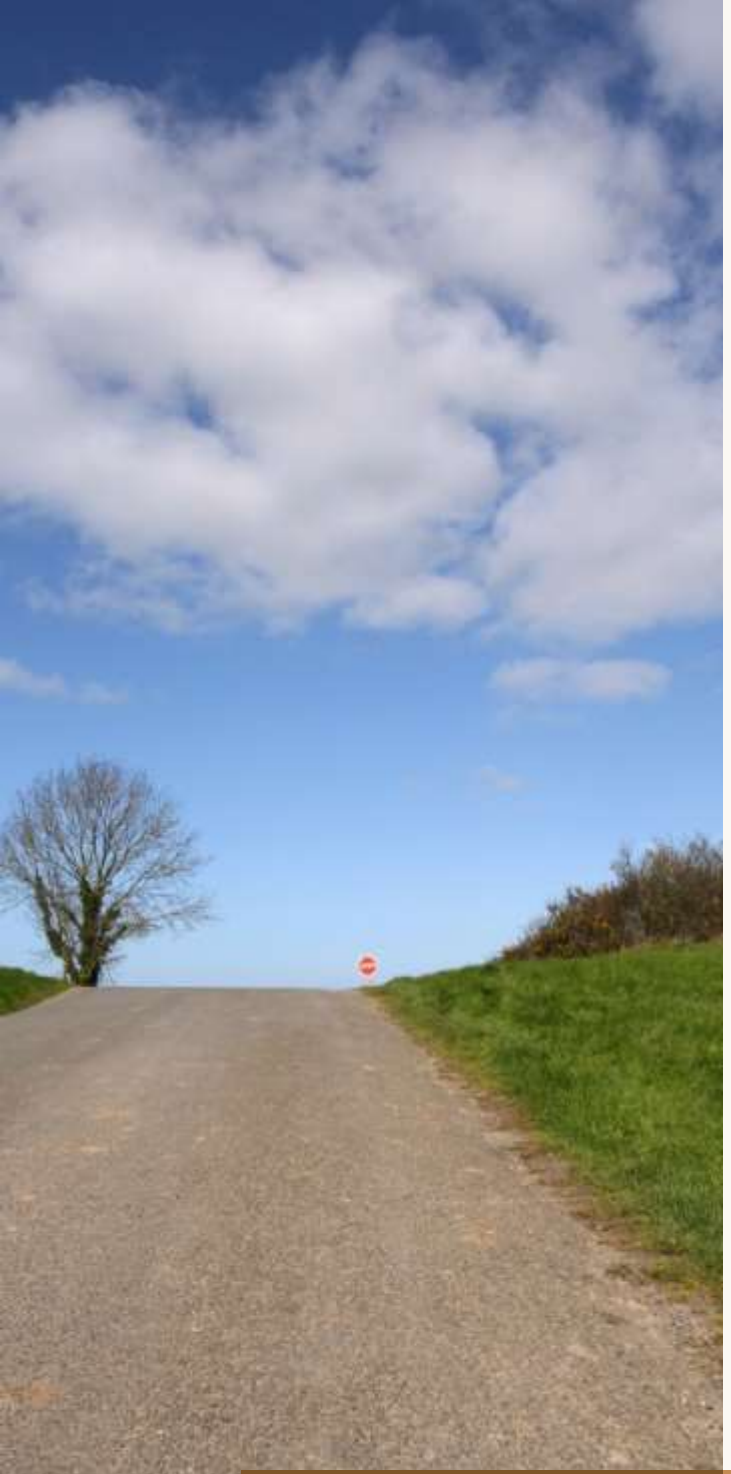
Для первой версии изменений — годится (но сделать можно ещё очень многое).

Как видим, восемь волшебных вопросов всего за несколько минут позволяют составить солидный список желаемых улучшений — что, собственно, и требуется для *начала* дизайна. Берусь утверждать, что ни один другой метод не обеспечивает столь высокого КПД.

К сожалению, здесь есть интеллектуальная ловушка. То, что вы теперь знаете эти вопросы, ничему не помогает и ничему не способствует. Их знания недостаточно — их нужно задавать. Если вы не будете их себе задавать — вопросы никак вам не помогут. Чтобы это знание заработало, его нужно активно внедрять в свою проектную деятельность (как, собственно говоря, и любое другое знание).

А чтобы вам легче было их задавать, выполните несколько заданий для самопроверки. Самостоятельно пройдите весь цикл анализа для:

- Двух любых браузеров, например для IE и Opera.
- Двух любых программ просмотра и каталогизации изображений, например для ACDSee и Picasa.
- Двух любых информационных сайтов одной направленности, например для Gazeta.ru и Lenta.ru.



Заключение

Поздравляю — вы наконец-то дошли до этой страницы (что, впрочем, было не особо сложно, учитывая микроскопический объем книги). Пришло время подвести итоги. Уступая своей страсти к спискам, ещё раз перечислю основные тезисы этой книги:

- Простейший способ загубить проект — работать неэффективно, т. е. делать что-то нужное, но делать это слишком медленно, проблемно и т. п. Профессионал тем и отличается от любителя, что он должен как можно полнее и планомерней повышать продуктивность своей работы.
- Помимо эффективности, важно ещё и адекватное целеполагание. Проектная деятельность велика и обширна, очень легко сделать не то, что нужно. К счастью, избежать этого тоже легко — нужно всего лишь решить, к какому именно интерфейсу надо стремиться, — а затем проверить полученный интерфейс на соответствие задуманному.
- Как неэффективность работы, так и неадекватное целеполагание могут быть просто исправлены посредством нехитрой техники — методичного задавания себе вопросов о достигнутом на данный момент результате работы.

Безусловно, нужны ещё и опыт, способности, специальные знания, вдохновение или просто удача. Но эти темы либо в принципе не

книжные, либо не помещаются в одну работу (поэтому я охотно отсылаю вас к другим книгам; см. следующую страницу).

Спасибо за внимание.

Конец.

Рекомендуемая литература

Ниже приведены основные книги на русском языке, способствующие росту навыков дизайнера интерфейсов.

Я ограничил список только книгами, вышедшими в последние пять лет, поскольку шансы приобрести более старые издания очень невелики. Кроме того, я не включал сюда издания, выполненные слишком уж непрофессионально (напечатанные на плохой бумаге, с опечатками и т. п.), поскольку не хотел рекламировать книги, не стоящие своих денег.

Хочу, впрочем, заметить, что профессионалу, на мой взгляд, лучше всё-таки собирать коллекцию всей доступной ему литературы хотя бы для архива (преимущественно в оригинальных изданиях, поскольку качество работы отечественных издателей оставляет желать лучшего). Но начинать собирать свою библиотеку лучше с перечисленных мной книг.

Дизайн интерфейсов и смежные вопросы

Алан Купер об интерфейсе. Основы проектирования взаимодействия. Символ-Плюс, 2009. ISBN 978-5-93286-132-5
Пожалуй, самая главная книга о дизайне пользовательских интерфейсов.

Кристина Уодтке: Информационная архитектура. Чертежи для сайта. Кудиц-образ, 2004. ISBN 0-7357-1250-6
Чуть ли не единственная книга на русском языке, посвященная высокоуровневому проектированию интерфейсов веб-сайтов.

Якоб Нильсен, Хоа Лоранжер. Web-дизайн. Удобство использования Web-сайтов. Вильямс, 2007. ISBN 978-5-8459-1222-0
На основе большого набора юзабилити-тестов предметно, с многочисленными примерами, рассматриваются типичные ошибки в веб-дизайне.

Стив Круг. Веб-Дизайн: книга Стива Круга или “не заставляйте меня думать!”. Символ-Плюс, 2008. ISBN 978-5-93286-099-1.
Очень простая в изложении книга о том, насколько сложными (и сложными неоправданно) для пользователей могут быть многие варианты веб-дизайна.

Дженнифер Тидвелл. Разработка пользовательских интерфейсов. Питер, 2007. ISBN 978-5-91180-073-4
Подробно разбирается множество паттернов проектирования интерфейсов.

Вопросы языкознания

Джозеф Уильямс: Стиль. Десять уроков для начинающих авторов. Флинта, Наука, 2005. ISBN 5-89349-456-3
Адаптация одного из самых известных англоязычных учебников письма. В этот список книга попала, потому что дизайнеру вообще полезно уметь писать, а уж дизайнеру интерфейсов полезно тем более.

А. Э. Мильчин. Методика редактирования текста. Логос, 2005. ISBN 5-98704-033-7
Более подробный учебник редактуры.

Графический и информационный дизайн

Виктор Папанек. Дизайн для реального мира. Издатель Д. Аронов, 2004. ISBN 5-94056-007-5
Знаменитая книга о гуманитарных аспектах дизайна. Левацкое содержание, на мой взгляд, очень актуально в России, стране оголтелого консумеризма.

Владимир Лаптев. Модульные сетки. Проектирование многополосных изданий. РИП-холдинг, 2007. ISBN 5-903190-13-8
Модульные сетки как способ организации объектов на плоскости полезны не только дизайнерам-полиграфистам, но и дизайнерам интерфейсов — и те и другие решают одинаковые задачи по размещению контента.

Дж. Фелличи. Типографика: шрифт, верстка, дизайн. БХВ-Петербург, 2004. ISBN 5-94157-345-6
Полезный современный учебник верстки. Почти все правила можно и нужно применять и в дизайне интерфейсов.

Тимоти Самара. Структура дизайна. Стильное руководство. РИП-холдинг, 2008. ISBN 978-5-903190-27-0
Современный и актуальный учебник графического дизайна.

Маркетинг

Джеральд Залтман. Как мыслят потребители. То, о чем не скажет потребитель, то, чего не знает ваш конкурент. Прайм-Еврознак, 2006. ISBN 5-93878-239-2
В книге довольно глубоко рассматриваются вопросы удовлетворенности потребителей; поскольку она тесно связана с удовлетворенностью пользователей, книга будет полезна любому дизайнеру интерфейсов.

Гарри Беквит. Продавая незримое. Руководство по современному маркетингу услуг. Альпина Бизнес Букс, 2007. ISBN 5-9614-0442-0
В книге прекрасно иллюстрируются проблемы продажи услуг, невидимые самому продавцу. Поскольку дизайн — это услуга, книга будет полезна любому практикующему дизайнеру.

Об авторе

Вместе с Александром Белышкиным основал Usethics, первую в России юзабилити-компанию.

По всей видимости, спроектировал больше интерфейсов, чем кто-либо другой в России (и руководил разработкой ещё большего числа интерфейсов).

Автор первой оригинальной русскоязычной книги о дизайне программных интерфейсов (более недоступна; это её вторая, полностью переработанная версия).

Обратная связь

Нашли ошибку или опечатку? У вас есть полезное дополнение? Не сочтите за труд написать об этом по адресу deus@exmachina.ru.

Прочли книгу и составили о ней мнение? Изложите его в своем блоге, дневнике и т. п.

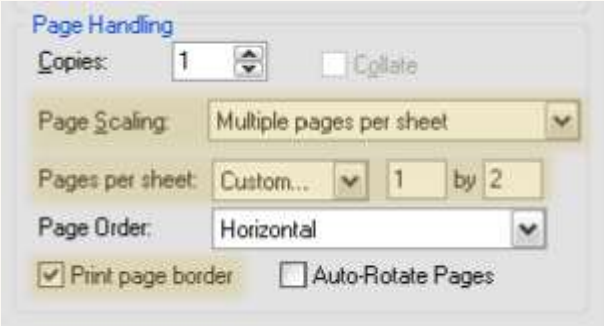
Заранее спасибо.

Колофон

Эта книга была написана в Q10, специальном текстовом редакторе для писателей, и сверстана в Adobe inDesign. В оформлении использованы гарнитуры Adobe Myriad и Minion Pro.

© Влад Головач
© Фотографии на обложке и на первых страницах разделов: David Ross, Chris Fourie, Ruslan Gilmanshin, Matdream, iphotoexpert, Millan, Laurent Renault, Avesun. Агентство — Dreamstime.com

Как читать эту книгу (с удобством)



Вы можете распечатать эту книгу. Рекомендую включить печать двух страниц на листе для экономии бумаги. Для этого в окне печати Acrobat (Ctrl+P для Windows, ⌘+P для MacOS) в списке *Page scaling* выберите *Multiple pages per sheet* и в появившихся полях ввода введите *1* и *2* (точно так, как показано на иллюстрации). Полезно будет так же включить печать краев страниц при помощи флажка *Print page border*.

Кстати, вы не потратите много тонера: фон и полоска снизу не напечатаются.

Вы можете читать эту книгу и с экрана. Рекомендую включить показ на весь экран (Ctrl+L для Windows, ⌘+L для MacOS). В любом случае, убедитесь, что у вас выставлен режим показа единственной страницы на экране (меню *View* → *Single page*) и включено сглаживание шрифтов (в настройках на странице *Display* надо выбрать нужное значение в списке *Smooth text*). Если буквы кажутся вам слишком крупными, не включайте полноэкранный режим, а уменьшите размер окна Acrobat. Страница уменьшится, соответственно уменьшатся и буквы.