

Содержание:

ВВЕДЕНИЕ

Актуальность исследования: Компьютерные программы могут создаваться в консольной версии или с привычным для пользователей Windows gui-интерфейсом. Интерфейс программы можно написать полностью самостоятельно, но гораздо проще создать его в той или иной среде программирования, используя визуальные компоненты.

Помимо того, что программы дают определенные команды компьютеру, эти команды спрятаны от наших глаз за красивыми (и не очень) кнопками, вкладками и т.д, то есть за интерфейсом - внешней оболочкой программы, тем как она выглядит, где расположены кнопки.

Для нас нажатие на кнопку — обычный щелчок мышки на кнопку, а для компьютера такое нажатие — целый ряд команд, интерфейс является своего рода переводчиком, нам он показывает, что можно делать и куда нажимать, а для компьютера преобразовывает простые действия (вроде щелчков мышки) в сложные ряды действий.

У многих программ возможно изменение пользовательского интерфейса (можно поменять внешний вид программы на свой вкус).

Интерфейс в компьютерной технике – это средство взаимодействия пользователя с программой, игрой или операционной системой самого устройства. Интерфейс позволяет узнавать любую программу или привычный текстовый редактор. Если пользователь отлично владеет навыком работы в графическом редакторе Paint, то он с легкостью освоит и другие подобные программы, так как они имеют похожий интерфейс. Другими словами, этот термин означает совокупность различных средств, которые помогают человеку управлять работой вычислительной машины.

Цель работы в изучении вариантов построения интерфейса программ: особенности и эволюция.

Предмет исследования – интерфейс программы

Объект исследования – эволюция интерфейса программы

Задачи работы:

Рассмотреть общее понятие об интерфейсе;

Перечислить варианты разработки интерфейса.

Методологическую основу исследования составил диалектико-материалистический метод познания объективной действительности. Использовались также специально-научные методы (исторический, логический, функциональный, системный, сравнительный) и частно-научные методы (формально-юридический и сравнительно-правовой).

ГЛАВА 1. ОБЩЕЕ ПОНЯТИЕ ОБ ИНТЕРФЕЙСЕ

1.1 Понятие программного интерфейса и его значение

Программный интерфейс (англ. programming interface) – это способ общения пользователя с программируемым устройством или прикладными программами, а также способ обмена информацией между самими программами. Он определяет функциональность и удобство такого общения посредством реализации оптимальных параметров программ.

Кроме того, его целью является минимизация усилий пользователя во время подготовки исходных данных, их обработки и последующей оценки результатов.[5]

Целевое предназначение

Программные интерфейсы могут предназначаться для одного или многих пользователей, а так же для решения одной или многих задач. Во время их создания учитываются цветопередача, навигация, эргономика, модульный дизайн и удобство управления. Помимо этого они должны позволять быстро и с минимальным количеством допущенных ошибок и потраченного времени решать поставленную задачу. В большинстве случаев для увеличения привлекательности среди обычных пользователей используются упрощенные варианты программных интерфейсов. Высокотехнологичные интерфейсы применяются в основном

специалистами для решения более сложных и узконаправленных задач.

К основным задачам интерфейса программы или операционной системы являются ввод и вывод информации, управление программным обеспечением, командные операции и обмен данными при помощи различных внешних носителей. Даже задняя панель системного блока компьютера также является интерфейсом, который позволяет подключать другие устройства. Интересным фактом является то, что панель управления на стиральной машине или в автомобиле также по своей сути является интерфейсом, что раскрывает масштабные значения этого понятия.

Интерфейсом программы именуют разные части, с использованием которых может производиться управление этой программой. В программе интерфейс выглядит как окошки, кнопочки, которые используются для того, чтобы программа могла совершить действия, которых вы от неё ждете. [4]

Приведем простой пример в использовании компьютерных программ. Для того чтобы посмотреть фильм, необходимо воспользоваться такой программой, как видеоплеер. В программе запускается та строка, которая обозначает фильм, после чего он отображается на экране. Программа для просмотра фильмов также имеет свой интерфейс, который служит для того, чтобы управлять. Так, с помощью кнопок, присутствующих на плеере, можно сделать звук громче или тише, поставить фильм на паузу или выполнить другие необходимые действия.

Графический интерфейс

Графический интерфейс – это пользовательский интерфейс, в котором используются изображения вместо цифр. Изображения в нем заменяют также и буквы, это кнопки или иконки. Яркий пример интерфейса графического типа – это рабочий стол Виндоуз. Работа в этом интерфейсе заключается в том, чтобы обеспечивать работу программы с помощью кликов.

По сравнению с тем, что происходит ввод и вывод через командную строку, графический интерфейс простой и понятный. Не так часто для того, чтобы пользоваться графическим интерфейсом, необходимы специальные знания компьютера. Графический интерфейс часто интуитивно понятен, еще его называют дружелюбным. [1]

Графический интерфейс имеет и свои недостатки, главный из которых – это большой объем памяти, который необходим для того, чтобы в графическом виде представлять себе программу. Но современные программы этот недостаток

преодолели, поскольку память современных компьютеров увеличивается с каждым новым выпуском. Но вместе с этим усложняется и сам интерфейс, теперь он занимает больше памяти, но становится удобнее и эффективнее.

Что касается игр, то в них также существует графический интерфейс, чтобы пользователь мог взаимодействовать с компьютером во время игры. Он обеспечивает также и общение пользователей друг с другом. Почти во всех играх интерфейс сложный, и он позволяет управлять игрой с помощью кнопок и мышкой.

Действия игровых персонажей обеспечиваются действиями пользователя, и способы реализации их стандартны для всех практически игр. Часто пользователю предоставляется возможность поменять настройки интерфейса для того, чтобы ему удобнее было играть. Сейчас появились и новые возможности управления, так, при создании сенсорных экранов, управлять игрой можно с помощью прикосновения пальцев руки.

Помимо того, что интерфейс существует игровой, программный и графический, бывает также интерфейс следующих видов:

внешний;

внутренний.

Внутренний интерфейс представляет собой методы и свойства, к ним доступ осуществляется посредством иных средств данного объекта. Еще их именуют приватными. [11]

Внешним интерфейсом именуют методы и свойства, которые снаружи доступны пользователям. Такие методы называют публичными. Данные виды можно рассмотреть наглядно, взяв за пример кофеварку. Внутри кофеварки спрятан кипятильник, элемент, который нагревает, тепловой предохранитель и так далее. Все это можно назвать внутренним интерфейсом. Детали, которые его составляют, обеспечивают работоспособность прибора. Для этого они взаимодействуют друг с другом. К примеру, для работы кофеварки её нагревательный элемент подключен к кипятильнику.

Ко внутреннему интерфейсу кофеварки подобраться тяжело, он закрыт от пользователя пластмассовым корпусом. Детали прибора скрыты, и пользователю доступен только внешний интерфейс. Когда приобретена кофеварка, то пользователю доступен только внешний интерфейс. Знать о внутреннем

интерфейсе совсем не обязательно, для пользования прибором необходим только внешний его интерфейс.

Такие же примеры касаются и остальных бытовых приборов, к примеру, стиральная машинка, телевизор и прочее. Существует внутренний интерфейс и у компьютера, он не доступен пользователю, однако при поломке прибора, приходится взаимодействовать именно с ним.

Таким образом, интерфейс характеризуют как средства, с помощью которых удастся взаимодействовать с вычислительными машинами, управлять бытовыми приборами и так далее. Он бывает внешним и внутренним. Пользователю доступен только внешний интерфейс приборов и машин.

1.2. Этапы разработки интерфейса программы

Создание сложного и хорошего интерфейса — это процесс, объединяющий специалистов в различных областях. Он занимает продолжительное время, поэтому разработку интерфейса сайта или программы делят на определенные этапы.

Предложенное деление не является универсальным. Каждый из этапов можно поделить на подэтапы. И на подподэтапы — так процесс выглядит еще сложнее, а значит дороже в глазах клиентов :

1. Сбор данных

Сбор данных нужен для того, чтобы четко понять, что за продукт существует на данный момент, что в нем не устраивает клиента и какого результата он ждет от совместной работы. На этом этапе разработки интерфейса программы дизайнер:

общается с клиентом, чтобы понять смысл и философию программы;

смотрит наработки: готовые прототипы (пусть даже они существуют только на салфетке);

анализирует программы конкурентов (и, возможно, проводит тестирование юзабилити программ конкурентов);

проводит структурированные интервью с клиентами или потенциальными клиентами. [6]

2. Проектирование

Этап проектирования позволяет создать тестовую версию приложения, т.е. распределить по экранам функционал всего приложения. На этапе проектирования дизайнер интерфейсов:

определяет, на каком месте будут находиться та или иная кнопка и текст;

продумывает, как сгруппировать элементы;

определяет микротекст и поведение элементов.

Как результат, вы получаете динамичный прототип вашего приложения, который можно использовать для тестирования эргономичности или начала разработки приложения.

В прототипе содержатся все необходимые комментарии по поведению контролов, крайним случаям и т.д. Они похожи на комментарии к ячейкам "Эксель": нажал на иконку и прочитал краткий комментарий.

Опыт наших клиентов говорит, что это намного удобнее, чем работа с отдельным файлом Word или печатной документацией, поскольку не нужно переключаться между экраном и файлом. В результате работать приятнее: растет мотивация, скорость и качество кода.

Но мы также можем экспортировать прототип в виде файла Word с комментариями, это несложно и, естественно, входит в цену.

3. Дизайн

Если бы люди были роботами, то можно было бы остановиться на прототипе, но мы не роботы, поэтому для нас важна эстетика. На этапе разработки дизайна мы делаем интерфейс не только удобным, но и красивым.

На этом этапе разработки интерфейса программы дизайнер:

определяет сетку, цвета, шрифты и фон;

а также часто создает нестандартные элементы управления, такие как выпадающие меню.

Естественно, на каждом из этапов идет обсуждение и, при необходимости, бесплатная доработка. Ваш заказ вы получите либо в качестве графических файлов в формате Photoshop, либо в виде HTML- или XAML-кода. [7]

4. Имплементация

Когда с интерфейсом все понятно, дело остается за немногим :) Как правило, наши клиенты держат штатных программистов, а нас привлекают для различных работ, связанных с пользовательским интерфейсом, от проектирования до создания иконок. Однако, тем клиентам, у которых нет собственных разработчиков, мы предлагаем разработку и тестирование веб-приложений и мобильных приложений под iOS. У нас есть постоянный отдел разработчиков и тестировщиков. Мы гарантируем: никакого фриланса.

На этапе имплементации идет разработка и тестирование (QA, не юзабилити) программы. Разработчикам будет однозначно понятно, как что-то делать, исходя из графических файлов (скетчей) и пояснений к ним. В обратном случае, мы дорисуем и допишем.

Конечно, разработка делится на свои этапы, но мы не будем их здесь расписывать для краткости.

5. Юзабилити-тестирование

Привлечение к проекту хорошего дизайнера интерфейсов не избавит от необходимости систематически проводить тесты юзабилити. Полагаться только на «гениального дизайнера интерфейсов» вредно по следующим причинам:

естественно нужно стараться привлечь к работе над проектом лучших дизайнеров интерфейса (например, VisualPharm :) Но, к сожалению, это не всегда возможно. Порой в вашем проекте принимают участие те люди, которых вы можете к нему привлечь, а не те, о работе с которыми вы мечтаете;

дизайн не является точной наукой; даже если ваш дизайнер гений, не все его идеи одинаково хороши. Потому для уменьшения риска логичным будет подвергнуть все эти идеи проверке в реальных условиях с реальными пользователями.

(напоминаем, новые идеи можно проверить с минимальными затратами с помощью таких техник, как бумажный прототип);

каким образом дизайнеры интерфейсов вообще становятся хорошими дизайнерами? Очень просто: учась на опыте тому, какие идеи работают, а какие нет. Но для получения этого опыта необходимы тесты, которые и проводят специалисты по юзабилити;

даже самые лучшие дизайнеры могут создать успешный продукт только в том случае, если они решают правильно поставленную задачу. Замечательный интерфейс не поможет, если безграмотно выстроен функционал. А каким образом дизайнеры интерфейсов узнают, что необходимо пользователям? Ответ прост: с помощью юзабилити-исследований;

никто не идеален. Даже очень хороший дизайн может быть улучшен, если его пропустить через процесс поэтапного улучшения качества. На каждом этапе вы проводите тесты с пользователями и на основе результатов, шаг за шагом, улучшаете качество пользовательского интерфейса.

Как правило, мы рекомендуем клиентам проводить удаленное юзабилити-тестирование. Преимущества такого вида тестирования:

быстро: 2-7 дней на тест;

дешево — на один-два порядка дешевле, чем большие исследования;

в рамках вашей целевой аудитории.

Можно выполнять юзабилити-тестирование как прототипа, так и готового продукта. Общая рекомендация — вместо одного большого теста делать много маленьких итерационных тестов. Сделали, протестировали, поправили, протестировали снова. Это позволит найти и исправить неочевидные ошибки.

1.3 Эволюция интерфейса программы

С чего все начиналось. Пожалуй, сегодня не найдется такого пользователя, который испытал бы затруднения в работе с ОС версии 1.0, запущенной на первом Macintosh производства 1984 г., но большинство было бы поражено примитивностью этой системы. Несколько десятилетий назад компания Apple и во сне не помышляла о тех богатых функциональных возможностях, которые теперь считаются неотъемлемыми особенностями ее машин. В те времена пользователи не могли одновременно запускать две или более программы, включая системное

приложение Finder. [10]

Даже увеличение объема оперативной памяти не дало тесной интеграции приложений друг с другом и с Finder. Единственным инструментом управления ОС была программа Switcher. Она позволяла переключаться между прикладными программами, причем окно текущего приложения «соскальзывало» в левую сторону экрана, в то время как окно другой программы «выползло» справа. Таким образом пользователь получал доступ к любому из запущенных приложений, однако на экране по-прежнему могло отображаться только единственное окно.



Радикально изменилась эта ситуация в 1985 г., когда двое студентов из Беркли продемонстрировали в головном офисе компании Apple ПО MacWrite. В самом текстовом редакторе MacWrite не было ничего примечательного. Однако когда они уменьшили окно программы, заполнявшее весь экран, под ним неожиданно обнаружилось окно приложения Finder, запущенного в фоновом режиме. Это событие ознаменовало рождение новой, многозадачной Mac OS.

К чему мы пришли. После этого стремительного рывка вперед Mac OS постепенно обретала свои характерные особенности: от меню (иерархического либо всплывающего) и вкладок до перетаскивания — наиболее простого и эффективного метода взаимодействия человека и компьютера. И сейчас Apple опережает Microsoft в таких важных областях, как удаление приложений и управление памятью.



Mac OS X обещает далеко уйти от конкурентов и в другой сфере. Эта версия ОС должна стать надежнее предыдущих. Конечно, Apple и раньше предлагала инструменты, с помощью которых удавалось аварийно выгрузить «зависшие» приложения (например, путем нажатия соответствующих функциональных клавиш). Однако по-настоящему действенное «лечение» — это устранение причины, вызывающей сбой, а не борьба с его последствиями, что и было реализовано в Mac OS X. [8] Но богатые функциональные возможности — это палка о двух концах, тут и возникает закономерный вопрос: останутся ли компьютеры Macintosh под управлением новой ОС самими собой, сохранив свои прежние достоинства?

Что превращает Macintosh в Macintosh?

Пользователи Macintosh всегда вольны были перемещать и сохранять приложения и документы в своем «виртуальном» рабочем пространстве как им заблагорассудится. Другие системы, от Windows до Unix, обязывают запоминать местоположение объекта в иерархической структуре папок и файлов.

Теоретически это должно сводить к минимуму беспорядок на жестком диске, однако на практике беспорядок всего-навсего переносится с Рабочего стола машины в глубины памяти пользователя. Поскольку большинство из нас лучше справляются с неразберихой на Рабочем столе (который, по крайней мере можно увидеть), чем с кашей в собственной голове, преимущество Mac OS перед другими ОС налицо.

Dock вместо привычной панели внизу экрана. В первых, «черновых» вариантах Mac OS X Рабочий стол имел традиционный вид с располагающимися на Dock стандартными значками типа Trash (Корзины) и пр. Они были доступны по щелчку мыши в виде новых открытых диалоговых окон. Однако за последующие несколько месяцев Apple немало потрудились над совершенствованием своего детища, и уже в бета-версии новая ОС обрела ряд особенностей, отличающих Macintosh от всех прочих компьютеров.

В Mac OS X отсутствует характерная для Mac OS 9 панель со вкладками в нижней части экрана, содержащая пиктограммы. Ее заменил новый системный объект — Dock, на котором пользователь может располагать от двадцати до тридцати самых различных элементов. Это очень изящное решение, но большое количество помещенных на Dock одинаковых значков делает его существенно менее удобным для работы, чем обычная панель Mac OS 9.

Тут нелишне вспомнить: в свое время разработчики Apple пришли к выводу, что непоименованные пиктограммы вызывают у пользователей затруднения, и руководством к действию избрали лозунг: «Одно слово стоит тысячи картинок», — который, впрочем, и сейчас остается актуальным. К сожалению, названия значков, помещенных на Dock, не возникают до тех пор, пока не подведешь к ним мышью. И если у пользователя открыто шесть документов MS Word, то в поисках нужного ему придется подводить мышью поочередно к каждому значку, что долго и неудобно.

Internet на автономных компьютерах. Графический интерфейс пользователя Mac OS X нацелен на работу в Internet. Новый файловый браузер, по мысли Apple, призван «перенести» мощь Web-браузеров на Рабочий стол сетевых машин. К сожалению, глобальная компьютерная сеть и «локальный» Рабочий стол — это все-таки разные вещи. Ведь «посетитель» Сети путешествует среди миллионов Web-узлов, объединяющих миллиарды страниц, о содержании которых он ничего не знает. Операции с цифровыми данными на автономном ПК — совсем другое дело: здесь имеется сравнительно небольшое количество файлов, созданных (или переписанных из какого-либо источника) самим пользователем и организованных по хорошо известному ему принципу.

Возможно, новый файловый браузер найдет свою нишу в крупных локальных сетях, где необходимо отслеживать «судьбу» значительного числа документов, созданных (либо собранных) различными людьми. Однако, насколько известно, поклонники компьютеров Macintosh предпочитают организовывать свою жизнь по собственному усмотрению, а не так, как это задумали производители ОС. [11]

«Изюминки» индивидуальной настройки

Графический интерфейс пользователя Mac OS X не является чем-то «застывшим», раз и навсегда определенным: Apple предусмотрела возможность выполнять его настройку. Индивидуальные установки позволяют пользователю придать ОС любой внешний вид и выбрать наиболее удобный метод управления компьютером. Однако достаточно взглянуть на MS Windows, чтобы понять, к каким неудобствам может привести злоупотребление подобными вещами.

Создайте «свою» Mac OS. Конечно, владельцы Macintosh и раньше имели возможность варьировать системные настройки, но в жестко ограниченных пределах. Среди доступных средств оформления выделялись «кожицы» (skins) — специальные файлы, которые обеспечивали различные цветовые схемы Finder и панелей инструментов приложений. Одни «кожицы» были красивы, другие — отвратительны, но главное, они приносили в оформление Mac OS элемент эстетики. Некоторые инструменты, такие как QuickKeys и AppleScript, позволяли задавать и автоматизировать последовательность смены системных событий, но радикально изменить базовые установки ОС было нельзя.

Mac OS X подарила обладателям Macintosh, во-первых, долгожданную свободу выбора системных настроек, во-вторых, возможность с помощью огромного количества бесплатных дополнительных программ преобразовать ОС и приложения по собственному усмотрению. И если раньше «кожицы» только придавали Mac OS внешний вид последних версий Linux, то теперь Mac OS X обеспечит графический интерфейс пользователя Linux на вашем компьютере.

Переработанная Mac OS выглядит так, будто Apple решила полностью разрушить все то, что раньше считалось неотъемлемыми особенностями Macintosh. На самом же деле под этим скрывается задача радикального «исправления» недостатков ОС. Одни «исправления» изящны, но бесполезны, другие грозят перевернуть с ног на голову, казалось бы, раз и навсегда сложившиеся представления пользователей о работе с Macintosh. Некоторые вещи так же удивительны, как и новый Finder. И если Mac OS X завоеует признание потребителей, Apple продолжит совершенствование своего продукта в том же направлении и реализует в следующих версиях ОС самые передовые идеи.

Сохраняя простоту. И все же пользователей волнует, сохранит ли Mac OS X легкость эксплуатации, издавна присущую всем продуктам Apple? Возможно ли, чтобы столь мощная ОС была удобной и понятной для неспециалиста? Безусловно,

богатые функциональные возможности и простота графического интерфейса пользователя должны быть двумя сторонами одной медали. К примеру, компьютеры Palm Pilot удобны и любимы миллионами пользователей именно потому, что, имея относительно простой интерфейс, обеспечивающий выполнение узких задач, они в то же время сравнимы по производительности с «настоящими» Macintosh. [12]

Новые функциональные возможности Mac OS X превосходят все то, что мы видели на ПК когда-либо ранее. Однако способна ли Apple и дальше сохранять декларируемую простоту использования своих продуктов? Или она последует неудачному примеру Microsoft, Sun и других компаний, создающих громоздкие и неповоротливые «тракторы» для рядовых пользователей? Может быть, Mac OS X все-таки превратится в послушный в управлении и стремительный «Порш»?

В любом случае, по какому пути производители Macintosh ни пойдут, начало следующего года обещает стать знаменательным для всех пользователей этих машин.

ГЛАВА 2. ВАРИАНТЫ РАЗРАБОТКИ ИНТЕРФЕЙСА

2.1 Инструменты для разработки интерфейса

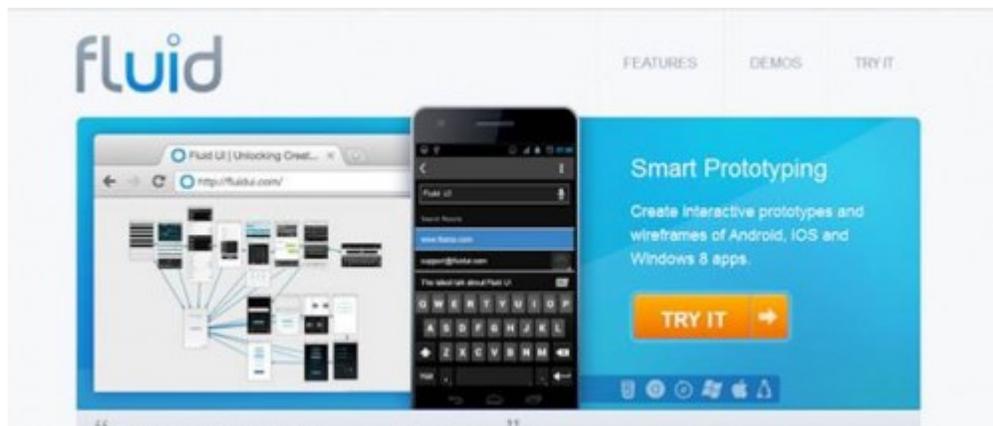
Прототипы стали незаменимым этапом профессиональной разработки веб-проектов и программного обеспечения. Прототипы облегчают жизнь всем людям, вовлеченным в процесс создания того или иного проекта: клиенту, менеджерам, дизайнерам и разработчикам. Прототип визуализирует конечный продукт и помогает выявить серьезные проблемы на самых ранних этапах, помогая избежать геммороя на поздних стадиях разработки.

Конечно, первоначальные эскизы, выполненные, к примеру, на бумаге и воплощающие первые идеи дизайнера еще никто не отменял. Но всю дальнейшую работу, в которую будут вовлечены все члены рабочей группы, лучше всего проводить в специальном программном обеспечении, которое поддерживает редактирование, управление версиями и совместную работу.

Сейчас мы рассмотрим 11 сервисов для прототипирования, часть из которых бесплатны, а другая часть — с весьма бюджетными для данной категории

Совсем недавно ребята представили премиум версию, в которой было добавлено несколько интересных особенностей, таких как неограниченное количество изменений макета и онлайн взаимодействие между членами рабочей группы.

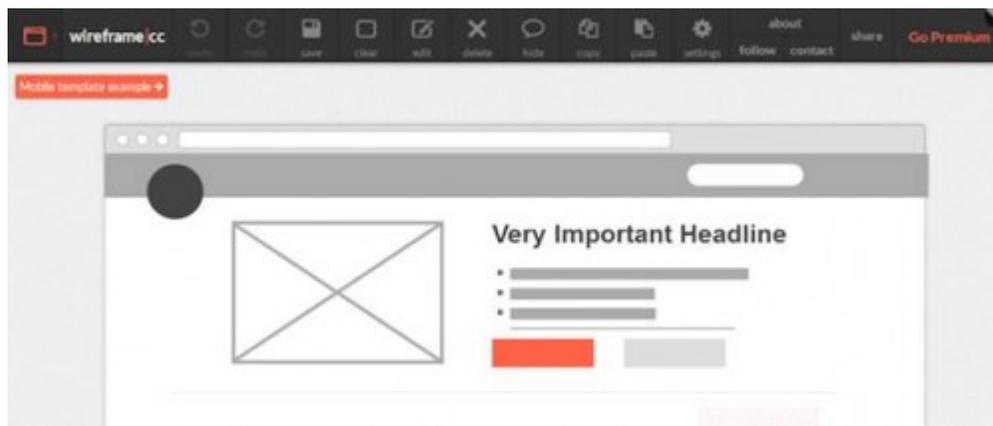
Fluid UI — средство разработки прототипов для iOS- , Android- и Windows приложений



Fluid UI — онлайн средство для прототипирования мобильных приложений. Может быть использован как для создания простых макетов, так и сложных высококачественных макетов и прототипов. Сервис содержит библиотеки с элементами интерфейсов для iOS, Android и Windows 8 приложений.

Аккаунт для одного проекта с ограничением в 10 экранов доступен бесплатно. Платные планы начинаются с \$29 в месяц.

Wireframe.cc — минималистичный бесплатный онлайн-сервис для прототипирования



Wireframe.cc отличается от других инструментов прототипирования чрезвычайно простым интерфейсом без огромного количества наворотов. Когда вы заходите на сайт сервиса wireframe.cc, вы сразу же можете начать рисовать, перетаскивая

элементы мышью по рабочей области. Чтобы сохранить макет достаточно нажать кнопку «Save», после чего сервис сгенерирует уникальный URL, которым вы сможете поделиться с другими людьми или добавить его в закладки. [8]

Wireframe.cc на данный момент бесплатен, премиум-версия с доп. функциями ожидается в ближайшее время.

Axure RP



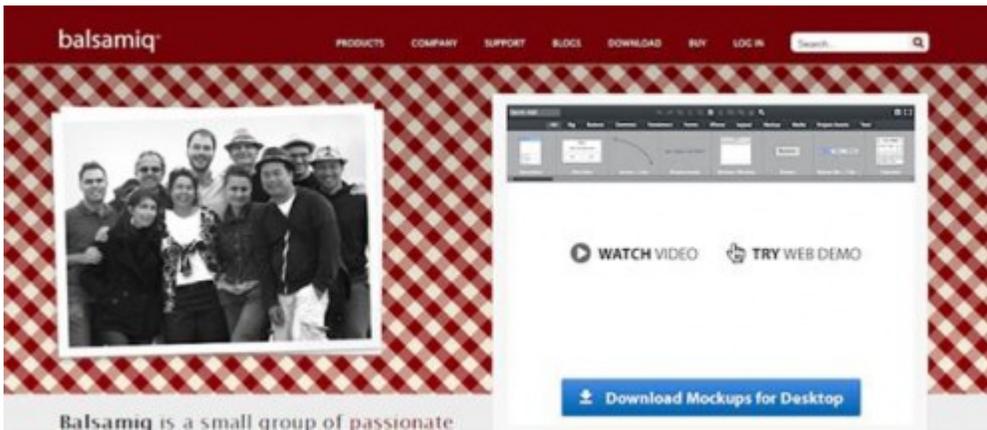
Цены на это ПО начинаются с \$289 за стандартную лицензию. Конечно, такая цена весьма велика. Однако, набор функций программы вполне оправдывает цену для некоторых категорий пользователей.

Axure может быть использован не только для создания простых макетов, но и для создания комплексных прототипов, а также для нормативных документов.

Из особенностей стоит также упомянуть возможность подключать к проекту клиента, чтобы показывать ему результат, отслеживать обратную связь, управлять дискуссией, удобно взаимодействовать с членами команды.

Axure — это десктопное ПО, доступное для Mac OS X и Windows.

Balsamiq Mockups — инструмент для быстрого прототипирования на основе Flash



Balsamiq Mockups — чуть ли не самый известный инструмент для прототипирования.

Продуктом, выпущенным в 2008 году силами одного программиста, в настоящее время пользуются десятки тысяч клиентов. Команда сервиса выросла до 11 штатных сотрудников.

Balsamiq Mockups стремится передать опыт рисования макетов на бумаге, поэтому прототипы выглядят не так формально, как в других программах.

Вы можете использовать Balsamiq Mockups непосредственно в браузере (потребуется Flash) или загрузить в качестве приложения Adobe AIR. Однопользовательская лицензия стоит \$79.

Pencil — инструмент для прототипирования с открытым кодом



Pencil — инструмент с открытым кодом для создания GUI-макетов и прототипов. Содержит большое количество встроенных фигур, начиная от элементов общего назначения и элементов блок-схем до элементов управления пользовательских интерфейсов (речь как о десктопных приложениях, так и о мобильных).

Вы также можете скачать пользовательские коллекции элементов из раздела Download на сайте Pencil. Инструмент доступен как автономная версия для Linux, Windows и Mac OS X. Также может быть использован в качестве расширения Firefox.

UXToolbox — инструмент для прототипирования под Windows



С помощью UXToolbox вы сможете создавать макеты и интерактивные прототипы для мобильных приложений, веб-сайтов и десктопного ПО.

Что интересно, программа позволяет в один клик переключаться между скетчем и попиксельно-выверенным прототипом.

После того, как вы закончите работу над проектом, вы сможете экспортировать макет в формат HTML, PNG, XML и Word или распечатать материалы для рабочей группы.

UXToolbox работает с Windows XP и с более поздними версиями и стоит £159 (около \$240) за копию.

Mockups.me — инструмент для создания пользовательских интерфейсов



Mockups.me очень похож на Balsamiq Mockups, но имеет ряд преимуществ. В частности у Mockups.me есть приложения для IOS и Android. Кроме того, Mockups.me включает в себя встроенные коммуникационные системы для сбора обратной связи через аннотации и комментарии.

Стоимость десктопной версии Mockups.me — \$49, версия для планшета доступна за \$19,99, а цены за веб-версию начинается от \$99 за год.

Live Wires — приложение для создания прототипов под iPad



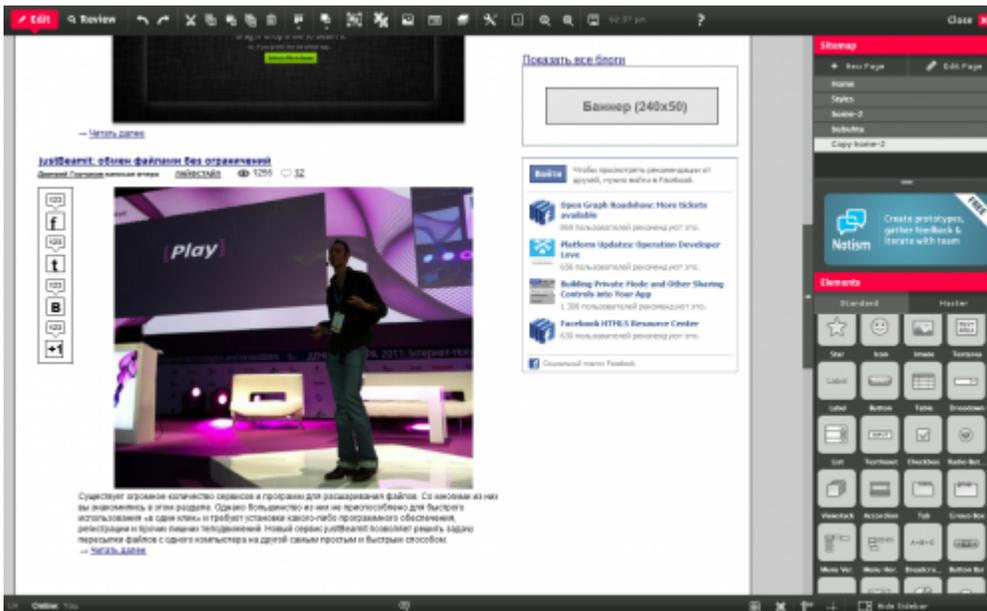
Live Wires поможет вам протестировать и создать прототипы непосредственно на iPad. Приложение может быть использовано для создания интерактивных прототипов приложений для iPhone и iPad.

Live Wires включает в себя большую коллекцию легко настраиваемых элементов управления.

Прототипы могут быть экспортированы по электронной почте а затем импортированы на другое устройство.

Live Wires сейчас доступен по специальной начальной цене в \$9,99.

HotGloo



Мой любимый инструмент для прототипирования. Ему посвящена отдельная статья.

Резюме

Как вы видите из приведенного выше списка, существует большой выбор доступных инструментов для прототипирования, из которых можно выбирать тот, который подходит именно вам.

2.2 Применение модульных сеток при построении пользовательского интерфейса сайта

Веб - пространство возникло относительно недавно и с этого момента интенсивно развивается и видоизменяется под воздействием множества факторов, появляются новые тенденции в дизайне, добавляются новые инструменты и технологии. Ежедневно создаются и размещаются в сети миллионы страниц. Но далеко не все страницы отличаются качественным дизайном, а также адаптацией к разнообразию разрешений экранов.

И если поначалу этими вопросами никто не задавался, то в настоящее время гармоничность композиции пользовательского интерфейса сайта и возможность его адаптации под различные разрешения экрана является ключевой проблемой современного веб - дизайна. Для решения этой задачи разрабатывается множество новых концепций, и подходов в верстке. Одним из способов решения данной

задачи является применение модульных сеток в веб - дизайне.

Проблемы создания соразмерной, гармоничной композиции в полиграфическом дизайне пытались решить еще в 20 - е годы XX века такие известные типографы и дизайнеры, как:

Эль Лисицкий, Вальтер Дексель, Ян Чихольд, Тео Бальмер, Макс Билл[4]. В 1961 году соотечественник Билла Йозеф Мюллер - Брокман впервые ввел в употребление термин модульный подхода к дизайну и опубликовал 28 примеров модульных сеток. В результате всех этих поисков в графическом дизайне появилась модульная сетка [4].

Однако если в полиграфическом дизайне модульные сетки используются почти столет, то в веб - дизайне этот метод начал применяться относительно недавно. Преимущество использования этого метода при создании пользовательских интерфейсов веб - приложений и сайтов в том, что он значительно упрощает проектирование и верстку макетов.

Приступая к проектированию пользовательского интерфейса сайта и созданию макета, грамотный веб - дизайнер для правильного построения каркаса проекта всегда пользуется вспомогательными инструментами. Модульная сетка – является мощной вспомогательной конструкцией, задающей единую систему пропорций и координат для расположения всех элементов дизайна, от заголовков и кнопок до блоков с формами и новостями.

Использование сетки в проектировании пользовательского интерфейса сайта дает возможность шаблонизирования, что позволяет создавать библиотеки типовых элементов и модулей сайта и последующего многократного применения в интерфейсе проекта [1].

Модульная сетка является мостом между композицией, выражающей идею дизайнера, и технической стороной реализации верстки макета. Модульная сетка позволяет дизайнеру управлять вниманием пользователя, а структурная и визуальная иерархия элементов обеспечивает быстрое и комфортное считывание информации. [2]. Пользователь бессознательно воспринимает дизайн интерфейса более гармоничным, упорядоченным и комфортным для восприятия.

Экспериментально доказано, что информация с четко и логично расположенными заголовками, иллюстрациями с подписями, текстами и подзаголовками не только быстрее и легче читается, а также лучше воспринимается и запоминается [4].

“Модульная сетка – это система построения визуальной информации на основе блоков – модулей, или система горизонтальных и вертикальных, а иногда и диагональных, дуговых и других кривых, направляющих, помогающих сориентировать и согласовать между собой отдельные элементы композиции” [5].

То есть, с помощью модульной сетки элементы сайта логически структурируются, тем самым облегая восприятие сайта и информации на нем.

Можно выделить 4 вида модульных сеток (рис. 1):

- а) блочная;
- б) колоночная;
- в) модульная;
- г) иерархическая [1; 3; 5].



Выбор вида модульной сетки в каждом случае зависит от сложности дизайна и требований к его гибкости, сетка с более маленьким шагом – более гибкая. Модульную сетку можно построить на начальном этапе проектирования макета и создавать дизайн на её основе, либо можно уже готовый макет выровнять по сетке [4].

Область применения модульной сетки в веб - дизайне безгранична. Уже сейчас модульные сетки используют при проектировании интерфейса сайта от простого сайта визитки, до корпоративного портала или международного новостного агентства. Так, например, на всех сайтах BBC используется одна универсальная сетка (рис. 2), которая является компонентом, недавно введенного в эксплуатацию, гайдлайна BBC «Global Experience Language».

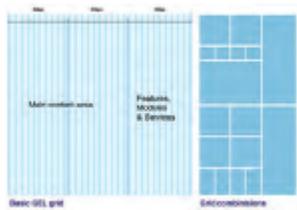


Рисунок 2 – Модульная сетка BBC

Как применить модульную сетку в своем проекте? Существуют универсальные модульные сетки для сайтов и веб - приложений, самые популярные из них:

Bootstrap;

Foundation;

Skeleton;

960grid.

В заключении стоит отметить, что модульная сетка это очень гибкий инструмент, позволяющий создавать новые и свежие идеи в дизайне шаблона сайта. Модульные сетки упорядочивают информацию в соответствии с её назначением и релевантностью, создавая композиционную структуру, как страницы, так и всего сайта.

С практической точки зрения модульные сетки ускоряют процесс разработки дизайна, за счет использования единой структуры. Новое содержание может быть легко добавлено без утомительной подгонки. И даже обновление дизайна может быть легко и быстро осуществимо, если содержание уже выровнено по сетке. Также использование модульной сетки позволяет создавать адаптивные сайты под экраны различных размеров и разрешений.

Модульные сетки дают возможность создания универсальных систем управления содержимым, с помощью которых обычные пользователи могут в кратчайшие сроки создавать хорошо организованные сайты, имеющие грамотно структурированную информацию на веб - странице и гармоничный дизайн, адаптированный под разные размеры экранов.

ЗАКЛЮЧЕНИЕ

Часто это слово мелькает в компьютерной терминологии, хотя частый гость и в совершенно ином контексте. В инженерной психологии термин объясняется, как разные методы общения между пользователем и оргтехникой. Обозначение «интерфейс» пришло от англичан, в переводе означает «между лицами». В области интернет-технологий этот термин охватывает унифицированные системы связи, гарантирующие обмен данными между объектами. Самый распространенный термин – «интерфейс пользователя» – набор способов, помогающих человеку управляться с аппаратурой.

Специалисты выделяют два вида:

Логический тип интерфейса. Комплекс установленных алгоритмов и договоров по обмену данными между элементами.

Физический тип интерфейса. Соединение автоматических, физиологических и многофункциональных данных, с поддержкой каковых связь реализуется.

Свою классификацию имеет этот термин в определении набора программных и технических средств, которые образуют взаимосвязь устройств:

Внутримашинный интерфейс – объединение проводов, схемы сопряжения с элементами ПК и алгоритмы передачи сигналов. Различают односвязные и многосвязные.

Внешний интерфейс – концепция взаимосвязи ПК с удаленными приборами. Есть интерфейс периферийных устройств и сетевой.

Интерфейс программы – набор руководящих составляющих программы, которые помогают юзеру совершать ряд действий: клавиши и окошки на мониторе. Чтобы увидеть кино, задействуют программу-медиаплеер, а уже регулируется изображение и звук кнопками и движками. Интерфейс системы гарантирует нужные данные в программах, выделяют два типа интерфейсных страниц:

Запросов, где внедрен подход, ориентированный на меню.

СПИСОК ЛИТЕРАТУРЫ

1. Агуров, Павел Интерфейс USB. Практика использования и программирования (+ CD-ROM) / Павел Агуров. - М.: БХВ-Петербург, 2006. - 564 с.
2. Информатика / Под ред. Н.В. Макаровой. - СПб.: Питер, 2012. - 160 с.
Информатика. Базовый курс / Под ред. С.В. Симоновича. - СПб.: Питер, 2015. -

640 с.

3. Информационные технологии в экономике управления / Под ред. В.В. Трофимова. - М.: Юрайт, 2015. - 482 с.
4. Кирсанов Д. Веб - дизайн: книга Дмитрия Кирсанова. - СПб: Символ - Плюс, 2013. - 368 с.
5. Мюллер - Брокманн Й. Модульные системы в графическом дизайне. Пособие для графиков, типографов и оформителей выставок / пер. с нем. Л. Якубсона. — М.: Изд - во
6. Студии Артемия Лебедева, 2014, -184 с.
7. Ляхович В.Ф., Молодцов В.А., Рыжикова Н.Б. Основы информатики. - М.: КноРус, 2016. - 348 с.
8. Михеева Е.В. Информационные технологии в профессиональной деятельности. - М.: Академия, 2013. - 384 с.
9. Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика. - М.: Академия, 2012. - 848 с.
10. Просветов Г.И. Анализ данных с помощью Excel. Задачи и решения. - М.: Альфа-Пресс, 2015. - 160 с.
11. Портянкин, Иван Swing. Эффективные пользовательские интерфейсы / Иван Портянкин. - М.: ЛОРИ, 2011. - 608 с.
12. Рассел, Джесси Интерфейс командной строки / Джесси Рассел. - М.: Книга по Требованию, 2012. - 665 с.
13. Эрглис, К.Э. Интерфейсы открытых систем. Учебный курс / К.Э. Эрглис. - М.: Горячая Линия Телеком, 2000. - 256 с.