

Тема 3

Разработка прототипа

1. Виды прототипов и технологии их создания
2. Основные компоновочные блоки макета страницы
3. Основные элементы графического пользовательского интерфейса
4. Инструменты для создания прототипов

1. Виды прототипов и технологии их создания ¹

«Разработка прототипа — средство, позволяющее проанализировать идеи, прежде чем потратить на них время и деньги. Все опытные мастера и инженеры создают образцы своих изделий до того, как начинают что-либо строить. Архитектор создает модель из бумаги или картона, либо с помощью виртуальных инструментов. Авиаинженеры используют аэродинамическую трубу. Строители мостов разрабатывают модели для исследования нагрузки. Разработчики ПО и веб-дизайнеры создают модели, имитирующие взаимодействие пользователя с их разработками.

Самая веская причина для создания прототипа — экономия времени и ресурсов. По сравнению с реальным продуктом прототипы просты и недороги в разработке. Таким образом, при минимальном вложении средств можно обнаружить ошибки создателей и юзабилити проблемы и улучшить пользовательский интерфейс до того, как сделаны значительные инвестиции в окончательную разработку и технологии.

Прототип страниц — это схематичное представление всех компонентов страницы и их взаимного расположения.

Прототипирование (создание прототипов) и итеративное улучшение проектов интерфейса признаются одними из наиболее мощных методов в проектировании взаимодействия человека и компьютера.

Прототипы используются для достижения одной или нескольких из следующих целей:

- проработать дизайн;
- построить общую коммуникационную платформу;

1. Брусенцова, Т. П. Проектирование интерфейсов пользователя: пособие для студентов специальности 1-47 01 02 «Дизайн электронных и веб-изданий» / Т. П. Брусенцова, Т. В. Кишкурно. — Минск БГТУ, 2019. — С. 95-99.

- увлечь других людей вашими идеями (например, руководство, других проектировщиков и т. д.);
- проверить техническую реализуемость;
- протестировать проектировочные идеи с помощью пользователей (клиентов).

В большинстве случаев прототип после тестирования оказывается неправильным, и его приходится переделывать, причем иногда полностью. Поэтому не следует чрезмерно наводить глянец и стремиться сделать его возможно более похожим на результирующую систему. Первый прототип стоит делать максимально примитивным. Только после того, как тестирование подтверждает его правильность, стоит делать более детализированный прототип.

В прототипировании выделяют четыре стадии.

Концептуальный дизайн — для исследования различных метафор в интерфейсе и подходов к дизайну.

Дизайн взаимодействия — для организации структуры экранов или страниц и переходов между ними.

Дизайн экранов — разработка внешнего вида каждого из конкретных экранов или страниц.

Тестирование — оценка прототипов для их последующего итерационного улучшения (при помощи небольшого тестирования с представителями реальных пользователей или с использованием «эвристического анализа» экспертами).

Виды прототипов и технологии их создания

Статичные прототипы: включают в себя такую популярную методику, как бумажное прототипирование, создание «скелетов» (набросков, эскизов интерфейсов — см. рис. 6.1), создание прототипов при помощи графических редакторов в виде статичных изображений (раскадровка).

Бумажный прототип. Необходимо нарисовать на бумаге все экраны и диалоговые окна. Нужно убедиться, что все интерфейсные элементы выглядят единообразно и сколько-нибудь похоже на реальные.

Рукописная схема поможет изучить наиболее важную часть приложения — контент. Понимание возможной схемы взаимодействия пользователя с контентом поможет дать более точную оценку числа страниц / экранов, необходимых в программе.

На первом прототипе вполне можно тестировать восприятие системы пользователем и ее основную логику.

Польза прототипирования на бумаге заключается, во-первых, в исключительной

простоте модификации по результатам тестирования, а во-вторых, в возможности безболезненно отлавливать представителей целевой аудитории.

Разумеется, значение слова «версия» весьма условно. В действительности после обнаружения каждой ошибки схема и прототип исправляются, а тестирование продолжается уже на новом прототипе. Так что на этом этапе прототип может пережить множество исправлений и, соответственно, много версий.

Прототип должен создаваться быстро. Его основная функция — показать часть функциональности в простейшем виде без акцентирования на иконках, цветах, шрифтах и тому подобном. Пример бумажного прототипа представлен на рис. 1.

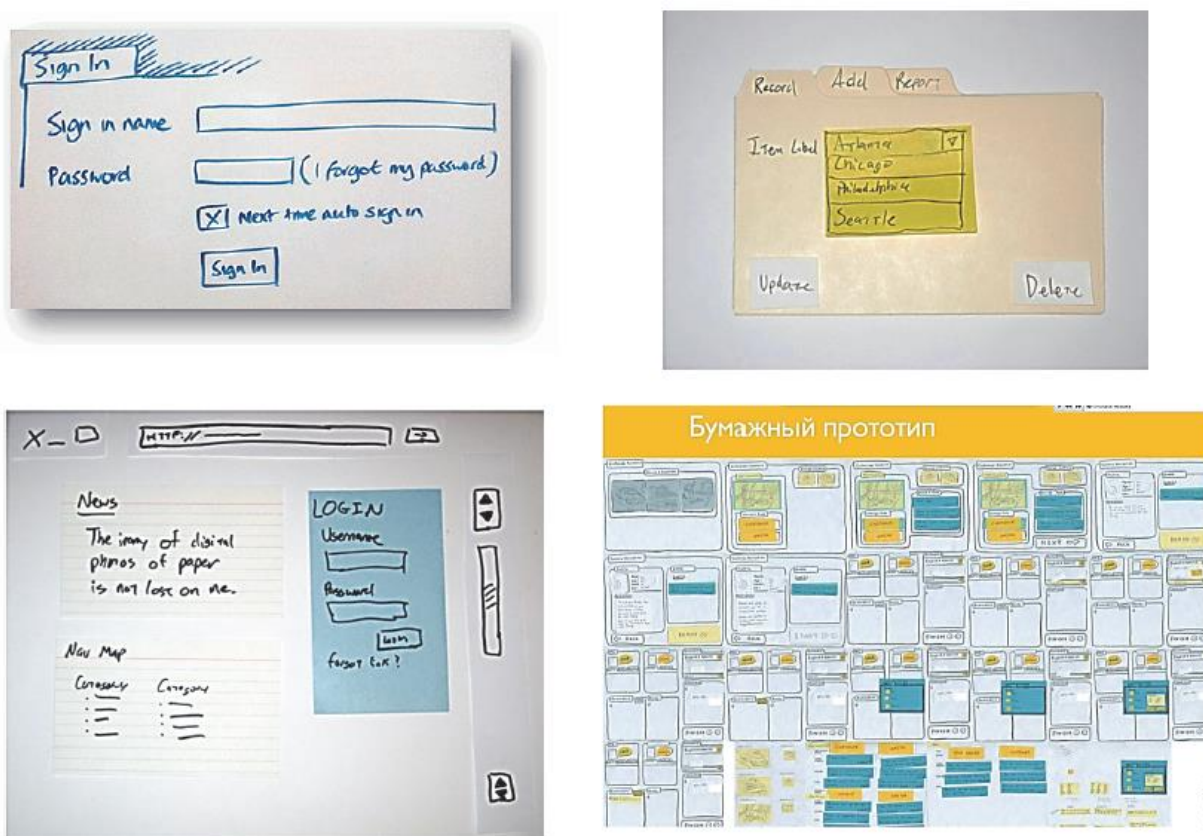


Рис. 1. Примеры бумажных прототипов

Раскадровка. Следующий шаг создание схемы для каждой страницы перехода. И здесь уже можно продолжать итерации, постепенно переходя от бумажных схем к цифровому прототипированию. Схема поможет быстро изучить возможные переходы на страницы / со страниц приложения. Скетчи позволят «оживить» приложение и понять большее количество деталей и структуры программы.

Это является промежуточным этапом между электронным и бумажным вариантами прототипов. Создаются при помощи средств электронного офиса. Для этого точно так же рисуется интерфейс, но уже не на бумаге, а в какой-либо программе (например, комбинации

Microsoft Visio и Microsoft PowerPoint). Данный вид решения определяется как статическая или пассивная раскадровка (рис. 2).

Здесь вы можете найти отличного ИТ-поставщика или поучаствовать в ИТ-тендере самостоятельно.

[Статьи и документы](#)
[Ответы на вопросы](#)
[Предложения по улучшению](#)

TenPro

[Зарегистрироваться](#)

Логин:
Пароль:
☐ запомните меня

[Не удаётся войти](#)

Я хочу объявить тендер

В чем состоит задача:

Конкурсная документация:

Я хочу участвовать в тендере

Искать тендер по ключевым словам:

[Расширенный поиск](#)

Последние компании-поставщики

- ООО «ПроСофтИнфоТехнологии»¹⁰⁰**
Разработка ПО, разработка веб-сайтов, системная интеграция
- ИТ-Солюшнс¹⁰⁰**
Разработка веб-решений, веб-маркетинг, раскрутка, поисковая оптимизация
- ЗАО «Мосгорпроект»¹⁰⁰**
Поддержка сетей, администрирование, поддержка ИТ.

Последние тендеры

- ЗАО «Сельхозмашморепродукт»⁷²⁸**
Разработка корпоративного веб-сайта
150 000 руб., 1 сентября — 31 декабря 2009
- ОАО «Мобильные инфотехнологии»³¹¹**
Разработка и интеграция CRM-системы
1 000 000 руб., 12—22 сентября 2009
- Российское общество охраны⁶³**
Подключение к сети интернет офиса

Рис. 2. Статическая (пассивная) раскадровка

Динамические прототипы. Дальнейшим развитием статической раскадровки является динамическая (интерактивная) раскадровка с применением средств анимации и т. п. При этом каждый экран получает отдельный слайд, а результат нажатия кнопок имитируется переходами между ними. Интерактивная раскадровка представляет собой электронный прототип.

На этом этапе создается действующая модель пользовательского интерфейса. Сделать интерактивный прототип можно быстро и дешево, особенно если использовать инструменты быстрого прототипирования. Получить его можно уже на раннем этапе проектирования, не дожидаясь отрисовки визуального дизайна, а значит, и начать юзабилити-тестирование.

При помощи программных инструментов (вроде Axure RP Pro, Microsoft Expression

Blend или плагина к MS Visio Intuitect) вы сможете точно показать, как интерактивные части сайта или приложения будут выглядеть для ваших пользователей (рис. 6.2).

С этой версией прототипа можно тестировать значительно более сложное взаимодействие человека с системой, нежели с бумажной. С другой стороны, исправление найденных ошибок значительно более трудоемко.

Существуют следующие подходы к прототипированию: черно-белый прототип на основе схем страниц (wireframe) и цветной прототип, основанный на принятом заказчиком визуальном дизайне системы.

Однако гораздо проще общаться с заказчиком, пользователями и разработчиками, имея на руках модель интерфейса, которая выглядит максимально похоже на финальный результат (рис. 3).



Рис. 3. Пример цветного интерактивного прототипа.»

2. Основные компоновочные блоки макета страницы ²

«К основным блокам можно отнести навигационные, информационные, сервисные, дизайнерские, рекламные (рис. 4).

Навигационный блок. Дизайн навигации кажется простым делом: нужно расставить на каждой странице ссылки, чтобы пользователь смог ориентироваться на ней. Однако если заглянуть чуть глубже, трудности навигационного дизайна станут очевидными.

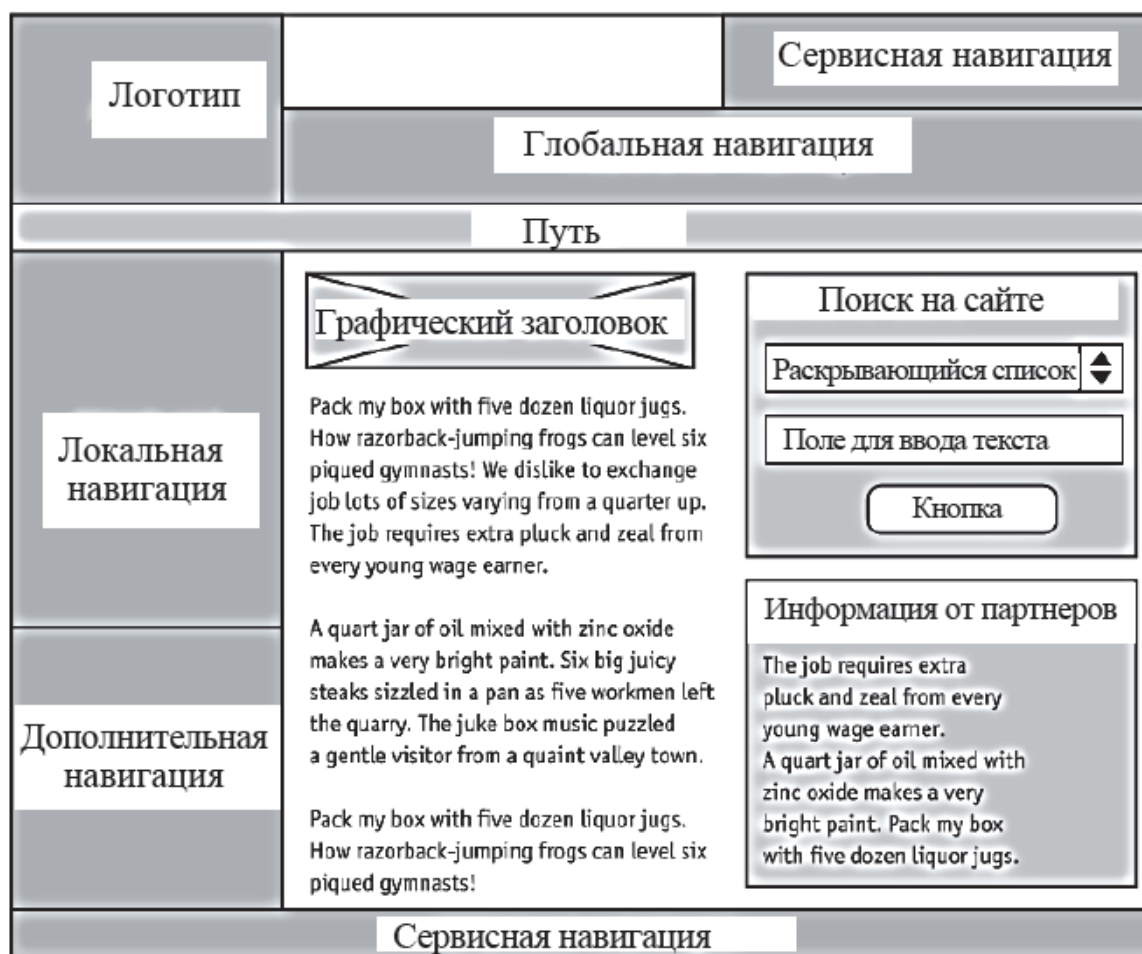


Рис. 4. Основные компоновочные блоки макета страницы

Дизайн навигации должен одновременно решать три задачи:

- *предоставлять пользователям способ попасть из одной точки страницы в другую.*

Поскольку во многих случаях связать каждую страницу со всеми остальными невозможно

2. Брусенцова, Т. П. Проектирование интерфейсов пользователя: пособие для студентов специальности 1-47 01 02 «Дизайн электронных и веб-изданий» / Т. П. Брусенцова, Т. В. Кишкурно. — Минск БГТУ, 2019. — С. 100-108.

(а если и возможно, то из общих соображений неразумно), приходится подбирать навигационные элементы так, чтобы они упрощали реальные передвижения пользователя; в числе прочего это подразумевает, что ссылки должны быть рабочими;

- *отражать взаимоотношения между внутренними элементами навигации.* Просто предоставить список ссылок недостаточно. Как эти ссылки соотносятся друг с другом? Являются ли одни более важными, чем другие? Какая между ними разница? Эта информация необходима пользователю для понимания того, какой у него есть выбор;

- *отражать связь между содержательной стороной элементов навигации и страницей, которая находится перед глазами пользователя.* Какое отношение имеет вся эта куча ссылок к странице, на которую я сейчас смотрю? Эта информация поможет пользователю понять, какой выбор ему следует сделать, чтобы наилучшим образом достичь своей цели или решить стоящую перед ним задачу.

Рассмотрим навигационные блоки.

1. «На главную страницу». Данный блок представляет собой гиперссылку, оформленную в виде логотипа либо названия компании. Он чаще всего ведет пользователя на главную страницу. Блок располагается обычно в левой верхней части страниц и должен быть повторен на всех страницах сайта, но на небольших персональных сайтах может и вовсе отсутствовать.

2. Поиск и быстрый переход. Несмотря на то что эти функции разные, их можно объединить в один блок по их внешней схожести и некоторой схожести по сути. Оба варианта представляют собой поле ввода (редактируемое или нет) с кнопкой выполнения введенного запроса. Наиболее часто блоки поиска и быстрого перехода располагаются в верхней правой части страницы. Они важны для навигации, поэтому «прятать» их от посетителя не рекомендуется, особенно если на страницах много информации.

3. Горизонтальное меню. Один из самых главных блоков на странице. Под блоком горизонтального меню понимается список гиперссылок, ведущих к основным разделам сайта. Гиперссылки располагаются на одной горизонтальной линии и могут быть оформлены и как обычный текст (меню), и как текст в виде изображения, и как символ (домик, корзина, конверт), и как вкладки. Они могут быть любого цвета и размера. Горизонтальное меню бывает двухуровневым.

4. Вертикальное меню. Блок вертикального меню по смыслу соответствует горизонтальному. В силу привычки данный блок располагают по левому краю страницы, но он может находиться и справа, и с обеих сторон одновременно. Этот блок бывает как статичным, так и с выпадающим меню или раскрывающейся древовидной структурой.

5. Вторичная навигация. Визуально представляет собой усеченный вариант

горизонтального или вертикального меню. С точки зрения информационного наполнения сайта этот блок не является главным и чаще всего содержит сведения о компании владельце сайта.

6. Навигация по выборке. Данный блок необходим при работе с выборкой некоторых объектов (изображения, ссылки, результаты поиска), которую невозможно отобразить целиком. Для перемещения между частями выборки используется специальная навигация. В целях удобства пользователя должен быть визуально выделен текущий фрагмент, а также должны отображаться соседние элементы. Если не хватает места для отображения всех порций, то следует использовать стрелки или надписи Следующая / Предыдущая.

7. Авторизация. Этот блок, наверное, самый предсказуемый и понятный. Он располагается в том месте, где пользователь должен идентифицировать себя, чтобы получить доступ к определенной информации, или чтобы система могла распознать его и основывать свою работу на ранее введенных данных (например, для очередного заказа в интернет-магазине).

8. «Подвал» (текстовые гиперссылки). «Подвал» используется для текстовых гиперссылок на основные разделы сайта. Он должен быстро загружаться и быть доступным в момент, когда фокус внимания находится внизу страницы. Достаточно удобно: закончил работу с этой информацией и готов перейти к следующей. Часто под «подвалом» понимают все, что находится внизу страницы, а не только список основных разделов.

9. Навигационная строка. Этот блок последовательность гиперссылок, определяющая путь посетителя к текущей странице. Навигационная строка показывает, где был пользователь, дает возможность быстро вернуться на один и более шагов назад. Если список элементов становится слишком длинным, то можно отображать только несколько первых и последних ссылок, а промежуточные заменить на многоточие.

Информационный блок. Часто обновляемую информацию удобнее всего размещать в виде информационных блоков. Такой способ публикации облегчает задачу добавления и обновления информации за счет структурирования данных, а также за счет использования возможностей импорта и экспорта.

Рассмотрим информационные блоки.

1. Содержание. Данный блок состоит из заголовка и одного или нескольких блоков с основным содержимым страницы.

2. Текущая информация. Этот блок используется при необходимости сообщить посетителю какие-то краткие сведения, полезные для его работы. Как правило, такие блоки маленьких размеров и располагаются относительно произвольно, часто внутри других блоков.

3. «Раздел». Данный блок достаточно обособленный и содержит информацию,

несколько отличающуюся от основного содержимого страницы. Это может быть анонс, новость, краткое описание услуг компании, опрос и т. д. В блоке должно быть название и содержание раздела, а может еще и гиперссылка, позволяющая перейти к полному содержанию, например, смотреть новость полностью. Содержимым блока может быть и текст, и набор гиперссылок, и изображение.

4. Изображения (галерея). Это набор изображений, совмещенный с блоком навигации по выборке, иногда может быть только одно изображение. Изображения не всегда загружаются достаточно быстро. Однако по непонятной причине на многих сайтах этот блок состоит из одного большого изображения, вместо нескольких маленьких, которые можно просмотреть в большем размере дополнительным шагом.

Лучше делать картинки двух размеров, т. е. делать два вида блока — для множества картинок и для одной. Если пользователь не хочет внимательно рассматривать каждую картинку из шести представленных, то он дождется загрузки шести маленьких изображений, а не вынужден будет ждать загрузки страницы с полномасштабным изображением, что в результате вызовет его раздражение.

Информационный дизайн играет важную роль и в тех задачах, где интерфейс должен не только получать какие-то сведения от пользователя, но и передавать ему информацию. Классическая задача информационного дизайна при создании успешных интерфейсов — сообщения об ошибках, еще одна — предоставление инструкций пользователю (задача непростая уже хотя бы потому, что труднее всего заставить их прочитать эти инструкции).

Каждый раз, когда система должна облегчить пользователю работу с интерфейсом путем предоставления информации (например, когда пользователь только начал работать с сайтом или совершил ошибку), это задача информационного дизайна.

Сервисный блок. Блок данных, в котором отображаются протокольные блоки.

Рассмотрим сервисные блоки.

1. «Выбор языка». Этот блок необходим, если ваш сайт поддерживает несколько языков. Часто его располагают рядом со вторичной навигацией, а наименование языка пишут на нем же самом.

2. «Пустой блок». Этот блок представляет собой пустое место между другими и может служить для отделения одного блока от другого, для рекламного баннера. Чаще всего он появляется сам по себе в процессе верстки (например, пустое место под горизонтальным меню, когда меню уже закончилось, а основное содержание страницы располагается ниже).

3. «Версия для печати». Наверное, наименьший по размерам блок, он инициирует вызов текущей страницы и оптимизирует ее для отправки на принтер средствами браузера. Как правило, он располагается вверху или внизу основного содержимого страницы и

совмещен с пиктограммой принтера.

Дизайнерский блок. Изображение, созданное дизайнером для украшения сайта и не являющееся одним из основных элементов содержания, может быть использовано и в качестве фона для иных блоков, например, для блока «Название и слоган» или блока текущей информации.

Рекламный блок. Несмотря на то, что такой блок напоминает «Раздел», он может выглядеть как угодно: тут все зависит от фантазии разработчика. Это может быть и мерцающий баннер на половину экрана, и маленькая «гиперссылка-завлекалка», и имитация кнопки, например, «Рейтингуется SpyLog» или «Участник Rambler Top 100». Реклама — она и есть реклама — всегда стремится к разнообразию.

К рекламным блокам можно отнести следующие.

1. «Название и слоган». Этот блок содержит название компании или самого сайта и, возможно, слоган вроде «Уважаем классику, ценим новое». Этот блок оформляется чаще всего крупным шрифтом, иногда на фоновом рисунке или вместе с логотипом компании. Назначение его, думаю, понятно — это громкое заявление компании о себе.

2. «Копирайт». Блок описания авторских прав компании-владельца сайта, иногда со ссылкой на его создателя, например «Дизайн SuperBe6-Studio».

Пример компоновки. Вариантов компоновки блоков великое множество, что позволяет создавать уникальные и запоминающиеся web-сайты. Прежде всего следует решить, к какой цели вы стремитесь: к выражению собственных творческих способностей или к популярному и удобному информационному ресурсу.

На рис. 5 приведен пример сайта инструментов для страховых компаний. В Ахиге были созданы макеты каждой страницы. Обсуждение их с клиентом помогло понять бухгалтерию страховой компании и разработать лучшее решение в кратчайшие сроки.



Рис. 5. Пример сайта инструментов для страховых компаний

На рис. .6 показан пример сайта PowerMockup, в котором предусмотрены шаблоны и инструменты для строительных каркасов прямо в PowerPoint. С выпуском версии 3 PowerMockup (Wireframing надстройка для Microsoft PowerPoint) был запущен и обновленный веб-сайт продукта. При разработке сайта сначала произведен мозговой штурм макета идеи на бумаге, прежде чем приступить к выполнению более детальных каркасов в PowerPoint.

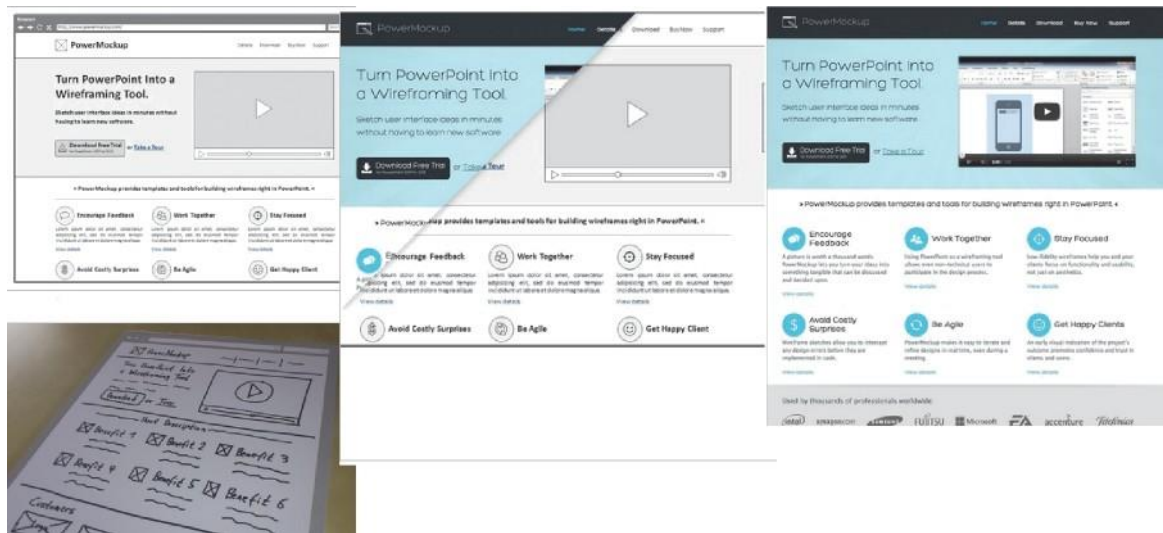


Рис. 6. Пример сайта PowerMockup

Проектирование каркасов UI для нового веб-сайта SkyPix Сент-Луис было первым важным шагом (рис. 7). Клиент хотел продемонстрировать много фотографий.

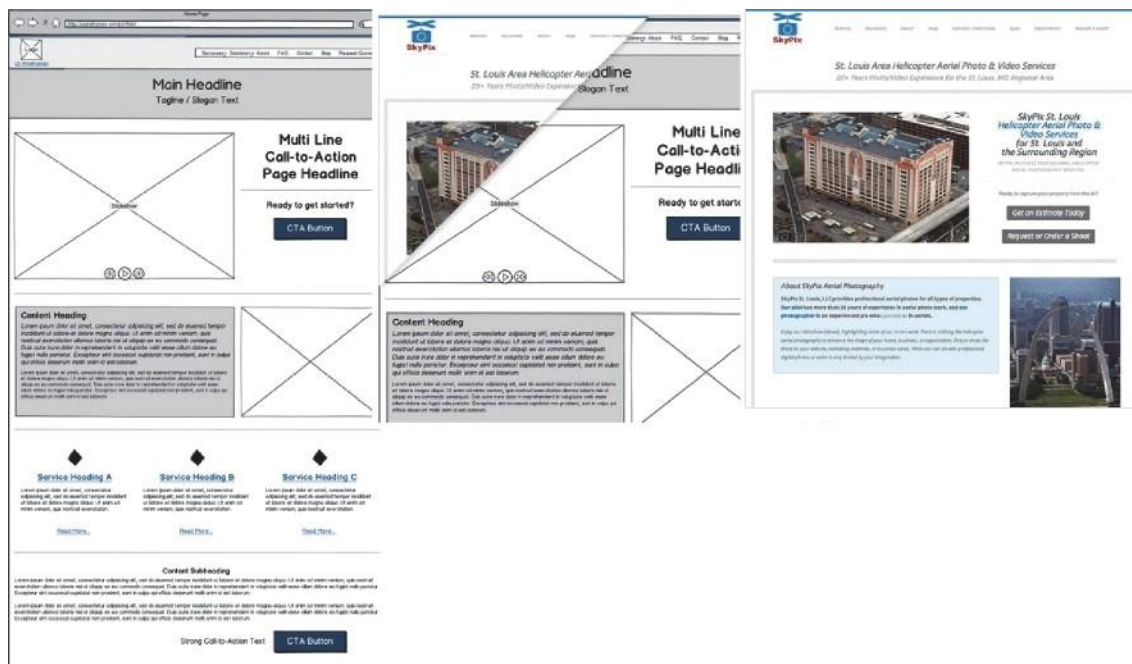


Рис. 7. Проектирование каркасов UI для нового веб-сайта SkyPix

Профессиональные каркасные макеты помогли продемонстрировать расположение и размер этих фотографий, а также другие различные графические элементы. В результате клиент смог лучше представить себе готовый сайт до начала стадии проектирования.»

3. Основные элементы графического пользовательского интерфейса

К основным элементам графического интерфейса относят окна, меню, линейки инструментов (инструментальные линейки), планки инструментов (tool bar), линейки прокрутки (scroll bar), кнопки (buttons), переключатели (radio buttons), наборы значений (value sets), выключатели (check box), списки (list box), текстовые зоны (text box) и т.п.

Рассмотрим их более подробно.

Окно – выделенная область экрана, визуально разграничивающая одновременно выполняемые процессы. Выделяют следующие типы окон:

1. главные окна программы,
2. окна документа,
3. режимные диалоговые окна,
4. безрежимные диалоговые окна,
5. палитры,
6. окна браузера (поскольку используемая в интернете технология существенно отличается от технологии ПО, этот тип окон выделяют в отдельную категорию).

Элементы окна

Окна имеют ряд общих элементов, главными из которых являются строки заголовка окна, строки статуса, панели инструментов и полосы прокрутки.

Строка заголовка. Текст и пиктограмма заголовка играют важную роль в программном обеспечении (они отвечают за переключение задач) и очень важную в интернете (обеспечивают навигацию). При проектировании ПИ по отношению к переключения задач действует простое правило «Релевантное выводится в первую очередь». Поскольку пользователю нужен именно конкретный документ конкретной программы, а вовсе не просто программа (окна документов, не попадающие в переключатель задач, нехороши), названия документов, как более релевантные, нужно выводить в первую очередь. Желательно сокращать длинные названия.

Строку заголовка можно использовать для вызова функций, которые нужны только наиболее опытным пользователям: нажатие на пиктограмму в строке заголовка вызывает

раскрывающееся меню. При этом меню для опытных пользователей будет отделено от меню общего назначения, что снижает вероятность вызова новичками (они могут даже не догадываться о существовании такого меню).

Также важно иметь в виду, что тот факт, что пользователи редко читают заголовки окна, не означает, что заголовки пользователям не нужны. Наоборот, хороший заголовок может облегчить понимание работы диалога. Поэтому наличие на экране заметного и адекватного заголовка окна является очень полезной составляющей ПИ.

Строка статуса может быть использована для отображения текущего состояния системы, и (или) для размещения панели инструментов для опытных пользователей. Рассмотрим эти функции подробнее.

Отображение текущего состояния системы. Практически каждая система имеет свойства, либо зависящие от документа, либо изменяющиеся со временем. Например, в иллюстративных программах объекты имеют какие-либо свойства, причем не все эти свойства показываются. Другой пример: когда система долгое время занята, она должна показывать пользователю индикатор степени выполнения. И, наконец, самый простой пример: пользователь текстового процессора имеет право знать, на какой странице документа он сейчас находится. Эффективнее всего выводить всё это в строке статуса, в первую очередь это относится к индикатору степени выполнения.

Панель инструментов для опытных пользователей. Зачастую система обладает функциональностью, которая с одной стороны важна, а с другой – сложна для начинающего пользователя. Обычно это касается не функций, а режимов работы системы. В таких случаях строка состояния позволяет, с одной стороны, сделать переключатели режимов непохожими на поля вывода (например, что метки ЗАП, ИСПР, ВДЛ и ЗАМ в статусной строке некоторых версий MS Word), что позволяет снизить вероятность ошибочного переключения. С другой стороны, если пользователь случайно щелкнет на переключателе, он сразу увидит изменение его вида и впоследствии, скорее всего, сможет переключиться назад. А опытный пользователь сможет переключаться между режимами так же легко, как и через панель инструментов.

Полоса прокрутки – элемент окна, осуществляющий перелистывание его содержимого. С появлением мыши со скрольным колесиком необходимость в полосе прокрутки значительно снизилась. При создании полосы прокрутки необходимо учитывать следующие рекомендации.

1. Размер ползунка должен показывать общий объем пролистываемого документа.
2. Ряд специалистов по прототипированию считают, что стрелки на полосах должны быть спаренными, т.е. обе стрелки должны находиться рядом, а не на разных сторонах

полоски. Это не очень логично, но более эффективно. Потом что если при перелистывании была допущена ошибка, спаренные кнопки позволяют минимизировать перемещение курсора к стрелке, ведущей в обратную сторону.

3. Если невозможно сделать динамическое изменение области просмотра при пролистывании, необходимо показывать текущее местоположение пользователя во всплывающей подсказке. При этом местоположение подсказки при перемещении курсора должно оставаться неизменным.

4. Необходимо обеспечить обработку погрешности перемещения курсора. Когда пользователь курсором перемещает ползунок, а смотрит в это время на документ, курсор может сойти с полосы. До определённого момента (смещение на 30-70 пикселей) система должна такое смещение игнорировать.

Панели инструментов представляют собой наборы пиктограмм, выбор которых инициирует какое-либо действие. Использование панелей инструментов в ПИ даёт следующие преимущества:

- они позволяют пользователям быстро вызывать нужные функции мышью;
- они позволяют пользователям меньше задействовать память;
- они повышают визуальное богатство интерфейса;
- они ускоряют обучение работе с системой (по сравнению с раскрывающимся меню) благодаря своей большей наглядности.

Однако они имеют и недостаток: занимают много места на экране, так что поместить в них всё, что хочется, невозможно. Решить эту проблему можно двумя способами. Во-первых, можно поместить в панель только наиболее часто используемые команды (как вариант, предусмотреть возможность индивидуальной настройки панели пользователем). Во-вторых, панель можно сделать зависимой от контекста действий пользователя. Можно использовать оба способа, т.к. они не противоречат друг другу. Панель инструментов как средство вызова функций позволяет экономить время, уходящее на открытие и закрытие диалоговых окон, и повышать интегральное качество взаимодействия с системой. В то же время панель инструментов нежелательно делать единственным способом вызова функции.

Изображения (Иконки). В интерфейсе непосредственного манипулирования пользователи выполняют действия непосредственно на видимых объектах. Этими объектами могут быть кнопки, метки, меню или изображения (иконки). Все иконки можно классифицировать согласно тому, насколько точно они отображают запускаемую ими функцию.

- Иконки подобия – иконки похожи на объекты, которые они отображают (типа ножниц, чтобы отобразить операцию «вырезки»).

- Иконки по образцу представляют пример типа объекта (например, иконкой, показывающей линию, чтобы представить средство рисования);
- Символические иконки используются, чтобы представить действие или состояние в символической форме (например, разорванная линия между двумя компьютерами для того, чтобы показать разорванное сетевое соединение);
- Произвольные иконки – не несут никакой информации по поводу их представления, поэтому их назначение должно быть описано (например, обратная круговая стрелка, чтобы представить действие «отмена последней команды»).

Курсоры. Курсоры являются хорошим средством обеспечения обратной связи. Например, каждый раз, когда пользователь подводит курсор к углу окна, курсор изменяется, показывая пользователю, что форму окна можно увеличить. Основное правило использования курсоров следующее: используйте индикацию видом курсора во всех случаях, когда вы можете сделать это эстетически привлекательно; в остальных случаях о курсоры неуместны.

Меню – это средство взаимодействия пользователя с системой, при котором пользователь выбирает из предложенных вариантов, а не предоставляет системе свою команду. Соответственно, диалоговое окно с несколькими кнопками (при отсутствии полей ввода) также является меню.

Существуют несколько различных классификаций меню, наиболее популярны две из них. Первая классификация делит меню на два типа:

1. Статические меню, т.е. меню, постоянно присутствующие на экране. Характерным примером такого типа меню является панель инструментов.
2. Динамические меню, в которых пользователь должен вызвать меню, чтобы выбрать какой-либо элемент. Примером является контекстное меню.

В некоторых ситуациях эти два типа меню могут сливаться в один, могут работать и как статические (пользователи нажимают на кнопки), и как динамические (пользователи вызывают меню) например, меню, состоящее из кнопок доступа к меню.

Вторая классификация также делит меню на два типа:

1. Меню, разворачивающиеся в пространстве (например, выпадающее меню). Всякий раз, когда пользователь выбирает элемент нижнего уровня, верхние элементы остаются видимыми.
2. Меню, разворачивающееся во времени. При использовании таких меню элементы верхнего уровня (или, понимая шире, уже пройденные элементы) по тем или иным причинам исчезают с экрана. Например, диалоговое окно с меню может перекрывать элемент управления, которым это меню было вызвано.

Статические меню из первой классификации, как правило, обеспечивают высокую

скорость работы, лучше обучают пользователей, но зато занимают место на экране. С динамическими меню ситуация обратная. Меню, разворачивающиеся в пространстве, обеспечивают большую поддержку контекста действий пользователей, но эта поддержка связана с потерей экранного пространства. Меню, разворачивающееся во времени бережно использует пространство, но зато хуже поддерживает контекст.

Меню содержит перечень команд, имеющихся в распоряжении пользователя при выполнении определенного шага задания или задания в целом. Меню дает возможность пользователю выбора необходимого средства решения задачи, не требуя от него запоминания имен команд и синтаксиса. Существуют различные типы меню:

- главное меню окна;
- выпадающие меню;
- контекстные (всплывающие) меню;
- каскадные меню.

Главное меню и выпадающие меню

Одна из наиболее распространенных форм меню – линейная последовательность команд (или разделов). Именно в таком виде выполнено главное меню окна, располагаемое горизонтально непосредственно под строкой заголовка программы. В связи с таким расположением главное меню иногда называют полосой меню.

В главном меню трудно перечислить все доступные команды. Поэтому в нем обычно располагаются только именованные разделы, содержащие группы команд определенного функционального назначения. Для отображения команд разделов используются выпадающие меню. Выпадающее меню отображается в виде столбца, размещаемого под выбранным разделом главного меню. В выпадающем меню команда, на которой размещается курсор выделяется цветом. После отпускания клавиши выбранная команда исполняется. Те операции, которые в данный момент времени не могут быть выполнены (например, нельзя удалить кадр, который не был создан) отображаются блеклым цветом.

Контекстные (всплывающие) меню

Контекстное меню дает пользователю доступ к операциям над объектами. Каждому объекту должно соответствовать свое контекстное меню, содержащее операции только для этого объекта. Контекстное меню отображается в отдельном окне, для его вызова используется щелчок правой кнопкой мыши по желаемому объекту. Контекстное меню выводится рядом с выбранным объектом, содержит столбец с возможными операциями над объектом. Выбранная курсором операция выделяется в меню цветом и при отпускании клавиши мыши исполняется, после чего контекстное меню пропадает.

Внешне контекстное меню похоже на выпадающее, но отличается от него тем, что включает только операции для конкретного объекта и для данной ситуации (в контекстном меню операции, которые в данный момент времени не могут быть выполнены, вообще не отображаются). Размеры контекстного меню обычно невелики.

При выборе порядка следования операций в контекстном меню нужно руководствоваться следующими рекомендациями.

1. Первыми должны располагаться основные команды (например, «Открыть»).
2. Во вторую группу должны быть включены команды, реализуемые через буфер обмена (например, «Вырезать», «Копировать», «Вставить»).
3. Последними должны идти команды редактирования дополнительных атрибутов (если они имеются).
4. Операции сходного функционального назначения следует объединять в категории, которые разделяются линиями.

Контекстные меню могут применяться к любым объектам: окна, пиктограммы и др.

Каскадные меню

Каскадное меню может использоваться для того, чтобы предоставить пользователю возможность дополнительного выбора и при этом не занимать дополнительного пространства в родительском меню. Каскадное меню содержит множество подменю, связанные с пунктами в родительском меню. Каскадное меню иначе называют иерархическим, а подменю в нем – дочерним. Если строка родительского меню имеет дочернее меню, то в правой части этой строки размещается признак наличия дочернего меню – стрелочка.

Каскадные меню усложняют интерфейс. Поэтому при их использовании нужно учитывать следующие рекомендации:

1. Минимизировать число уровней иерархии.
2. Избегать использования каскадных меню для доступа к часто используемым командам.
3. В качестве альтернативного решения применять вторичные окна и панели инструментов.

Заголовок меню

Каждое выпадающее и каскадное меню имеют собственные имя и текстовую надпись. Текстовая надпись информирует пользователя о назначении пункта. Имя связывает компоненты меню друг с другом и обработчиками событий. Имя для выпадающего меню – это имя пункта главного меню. Для каскадного меню это имя подпункта родительского меню, с которым связано каскадное меню. Заголовок должен обозначать предназначение меню. Желательно для него использовать одно слово, избегая сложных слов (например,

«Шрифторамер»). При этом следует помнить, что имя пункта и его текстовая надпись это не одно и то же.

Пункты меню

Каждый пункт меню соответствует определенному действию, который может выполнить пользователь в данной ситуации. Пункты в меню могут быть представлены в виде текста, графики (например, в виде пиктограммы), либо комбинации текста и графики. Всегда нужно визуально показывать, какие пункты меню сейчас недоступны, отображая их блеклым цветом. Если недоступны все пункты меню, то следует сделать недоступным и его заголовок.

Если пункт меню является командой, которая требует дополнительной информации, то название команды должно сопровождаться многоточием (...). При выборе такой команды должно вызываться дополнительное окно с запросом нужных данных. Справа от пункта может быть сокращенное обозначение клавиши быстрого доступа (горячая клавиша, клавиша-акселератор), при нажатии которой на клавиатуре обеспечивается быстрое исполнение команды. Например, Ctrl + C для копирования в буфер выделенного объекта. Использование таких клавиш избавляет пользователя от поиска нужной команды в меню. Горячие клавиши стоит применять только для часто выполняемых операций. При этом для стандартных действий нужно использовать общепринятые горячие клавиши. Например, во всех приложениях клавиша F1 применяется для вызова контекстной справки.

Пункты меню могут быть независимыми или взаимозависимыми. В них целесообразно вводить маркеры, размещаемые слева от имени пункта. Независимые пункты эквивалентны флажкам. Для визуального отображения установки таких пунктов рекомендуется использовать маркер флажка (в виде птички). Взаимозависимые пункты эквивалентны переключателям. Для визуального отображения выбора таких пунктов рекомендуется использовать маркер переключателя (в виде кружочка, с точкой при выбранном пункте).

Меню может содержать пункт, выбираемый по умолчанию. Для этого пункта должна поддерживаться техника ускоренного доступа (например, посредством двойного щелчка мыши). В соответствии с системными соглашениями такие пункты выделяются полужирным шрифтом. При выборе текстовых имен пунктов рекомендуются учитывать следующие правила.

1. Используйте уникальные имена в пределах одного меню.
2. Название пункта должно состоять из одного слова (глагол для действий, существительное для объектов).
3. Определяйте уникальные имена клавиш быстрого доступа (горячие клавиши) для часто выполняемых действий.

4. Придерживайтесь норм использования заглавных букв, принятых в языке.

Кнопкой называется элемент управления, позволяющий выбрать опцию или вызвать событие (например, запуск подпрограммы). Все взаимодействие пользователя с кнопкой ограничивается нажатием.

Командные кнопки (Push Buttons). Нажатие на командную кнопку запускает определенное действие (кнопки прямого действия). Эти кнопки изображаются в виде прямоугольника, в центре размещается короткое текстовое сообщение (слово), поясняющее, какое именно событие инициирует нажатие кнопки. Командная кнопка имеет три состояния: нормальное (кнопка доступна, но еще не активирована пользователем), нажатое (активированное) и недоступное (неактивное) (если недоступна функция, закрепленная за кнопкой, или если кнопка расположена в фоновом окне).

Чекбоксы (Checkboxes) и радиокнопки (Radio buttons) – кнопки отложенного действия, т.е. их нажатие не инициирует какое-либо немедленное действие, с их помощью пользователя вводят некоторые параметры, которые будут использованы после, когда действие будет запущено другими элементами управления. При этом радиокнопки являются кнопками единственного выбора, а чекбоксы – множественного. В группе радиокнопок как минимум одна кнопка должна быть проставлена по умолчанию. Радиокнопки отображают наборы состояний, они не инициируют действия. Эти наборы состояний должны быть статичны, т.е. не зависеть от контекста. Кнопки объединяются в группы, предпочтительно от двух до семи кнопок.

Списки (list boxes) – средство выбора требуемых объектов или их свойств. Списки представляют собой вариант реализации радиокнопок и чекбоксов. Списки – удобный и компактный элемент интерфейса, который занимает минимум места на экране и в то же время несет большую информационную нагрузку. Элементы списков могут быть представлены в текстовой или графической формах. Список должен обеспечивать визуализацию сделанного выбора и поддержку действий, связанных с выбранными пунктами.

Списки бывают пролистываемыми и раскрывающимися, при этом пролистываемые могут обеспечивать как единственный (аналогично группе радиокнопок), так и множественный выбор (чекбокс); а раскрывающиеся работают исключительно как радиокнопки. Скорость доступа к отдельным элементам и наглядность в списках принесены в жертву компактности (они экономят экранное пространство, что довольно актуально, особенно если количество элементов велико) и расширяемости (простота загрузки в списки динамически изменяемых элементов делает их очень удобными при разработке интерфейса, поскольку позволяет не показывать пользователю заведомо неработающие элементы).

Самым простым вариантом списка является раскрывающийся список. В раскрывающихся списках малая высота списка позволяет визуально отображать команды, собираемые из составляющих. Проклистуемые списки могут позволять пользователям совершать как единственный, так и множественный выбор.

Списки единственного (единичного) выбора (Single Selection ListBox) используется для выбора только одного пункта в списке. Список единственного выбора является промежуточным вариантом между группой радиокнопок и раскрывающимся списком. Он меньше группы радиокнопок с аналогичным числом элементов, но больше раскрывающегося списка. Соответственно, использовать его стоит только при необходимости экономии пространства экрана. Если содержимое списка не уместается в поле списка, то отображается линейка прокрутки, позволяющая просмотреть весь список. Списки множественного выбора позволяют выбрать несколько вариантов из набора предложенных.

Списки целесообразно использовать при большом количестве вариантов, либо когда перечень может изменяться. Порядок расположения элементов в списке зависит от содержания, он должен быть удобен пользователю. Для списка имен лучше применять сортировку по алфавиту, для дат лучше использовать хронологический порядок. Если в данной ситуации какой-либо элемент списка не может быть выбран, его не нужно включать в отображаемый пользователю список.

Ширина поля списка должна быть достаточной для отображения пункта средней длины. Если заранее невозможно подобрать такую ширину, то следует использовать один из следующих подходов.

1. Выберите ширину для размещения самого длинного пункта.
2. Используйте эллипсис (...) в середине или конце длинных названий, сохранив при этом символы, позволяющие понять пункт.
3. Включите в панель списка полосу горизонтальной прокрутки.
4. Если количество пунктов в списке превышает уместающиеся в поле списка, то включите в поле списка линейку вертикальной прокрутки.

Комбобоксы. Комбобоксами (combo box), называются гибриды списка с полем ввода: пользователь может выбрать существующий элемент, либо ввести свой. Комбобоксы бывают двух видов: раскрывающиеся и расширенные.

Поля ввода (edit text fields). Вместе с командными кнопками, чекбоксами и радиокнопками, поля ввода являются основой любого интерфейса.

Слайдеры (Sliders) или ползунки – элемент управления, состоящий из полосы прокрутки (slider bar), показывающей допустимо возможное значение изменяемого параметра (например, яркость или контрастность изображения), и индикатора (ползунка),

показывающего текущее состояние. Слайдеры могут быть горизонтальными или вертикальными, перемещение индикатора может быть квантованным или нет. Ползунки незаменимы, если пользователям надо дать возможность выбрать значение, стоящее в хорошо ранжируемом ряду, если значений в ряду много или нужно передать пользователям ранжируемость значений.

Крутилка (spinner, little arrow) – элемент управления, который позволяет пользователю уменьшать или увеличивать значение некоторой величины. Крутилки также могут содержать поле ввода. При этом, чтобы ввести значение в крутилку, пользователю не нужно убирать руку с мыши и переносить ее на клавиатуру (в отличие от обычного поля ввода). Перенос руки с места на место занимает сравнительно большое время (в среднем почти 0,5 с.) и сбивает фокус внимания. Кроме того, при вводе значения мышью система может позволить пользователям вводить только корректные данные и в корректном формате. Это уменьшает вероятность человеческой ошибки. Следовательно, использование крутилок для ввода любых численных значений оправдано.

4. Инструменты для создания прототипов

Существует много инструментов для создания прототипов и дизайна интерфейса. Рассмотрим некоторые из них.

Axure RP Pro — ориентирована на создание веб-сайтов. Генерирует кликабельный HTML-документ и документацию в формате MS Word.

Caretta GUN Design Studio специализированное средство, позволяющее создавать интерфейсы в разных визуальных стилях, аннотации к ним, раскладовки и т. п. Можно экспортировать прототип.

Balsamiq Mockups — подходит для быстрого создания макетов интерфейсов. Прототипы выглядят рисованными.

Adobe InDesign изначально рассчитан для верстки полиграфических материалов, тем не менее подходит и для отрисовки прототипов. На выходе: кликабельный PDF.

Adobe Fireworks — специально предназначен для прототипирования интерфейсов. На выходе — кликабельный PDF или HTML.

Adobe Dreamweaver предназначен для HTML-верстки. На выходе — кликабельный HTML.

Microsoft Expression Blend. Интегрируется с Visual Studio. Прототип можно преобразовать в конечный продукт. Использует технологии Silverlight или WPF.

Microsoft Silverlight — это программная платформа, включающая в себя модуль для браузера, который позволяет запускать приложения, содержащие анимацию, векторную графику и аудио- видеоролики, что характерно для RIA (Rich Internet application).