

## ГЛАВА 2



# Процесс прототипирования

Отсутствие дизайннерской студии	39
На что похож процесс прототипирования	39
Резюме	50

В проектировании архитектуры и продукта прототипирование — данность. Но это не обязательно верно для разработки программных продуктов.

Андерс Рэмси

**C**оздание прототипов — обычная практика во многих областях проектирования, например архитектуре или промышленном дизайне. Она не просто допустима, но ожидаема.

Почему бы не использовать ее и в разработке программного обеспечения? В конце концов, эта область имеет много общего с архитектурным и промышленным проектированием, в том числе следующие характеристики:

- Все эти процессы относятся к проектированию.
- Для коммуникации при разработке используются искусственно созданные объекты.
- Конечный результат — вещественный объект, который люди могут испытывать и использовать.

Я думаю, дело в первую очередь в том, что в создании программного обеспечения акцент часто ставится на разработке, а не проектировании (представители отрасли называют этот процесс не «проектированием», а «разработкой программ»).

При создании программного обеспечения о проектировании часто задумываются слишком поздно. Акцент делается на технологиях или функциональности. А в архитектурном или промышленном проекте — именно на проектировании. Форма следует за функциями.

Кроме того, разработка программ воспринимается как производственный процесс, а архитектурное и промышленное проектирование — как ремесло, умение.

Возможно, все дело в способе обучения в разных областях деятельности. Обучение компьютерным наукам сосредоточено на преподавании

технологий. В архитектурном и промышленном проектировании студентов учат принципам проектирования, включая то, что иногда называют *дизайнерской студией*.

## Отсутствие дизайннерской студии

В архитектурном и промышленном проектировании дизайннерская студия — процесс, а не просто некое пространство. Этот процесс преподается в каждой солидной программе обучения проектированию. Однако сложно найти дизайннерскую студию в компьютерной науке.

В студии вы создаете проект или прототип и показываете его товарищам по учебе. Они критикуют вашу работу, показывая ее сильные и слабые стороны.

Проектирование, представление результата и критику нужно воспринимать как совместный, быстрый, итеративный процесс. В ходе этого процесса вы делитесь идеями, успехами и неудачами со своими товарищами.

Дизайнерская студия — главный элемент архитектурного и промышленного проектирования. Прототипирование — главная часть обучения. Поэтому студенты рано осваивают этот навык и используют его регулярно. Прототипирование становится обычным инструментом в процессе проектирования.

## На что похож процесс прототипирования

Не существует общего процесса сверхпрототипирования, единого рецепта, как у кока-колы. Однако есть ряд опробованных на практике принципов, которым можно следовать независимо от того, над чем вы работаете.

Не важно, какой процесс прототипирования вы решите использовать. *Процесс* — это способ закончить работу. Одним из наиболее распространенных подводных камней оказывается чрезмерная вовлеченность. Если вы нацелены на выполнение процесса, а не на достижение конечной цели, вам не удастся добиться успеха.

## На что похож процесс прототипирования

К тому же в хорошем процессе соблюден баланс между структурой и гибкостью. Он обеспечивает твердую основу, в то же время давая возможность гибко вносить корректировки и адаптироваться к изменениям графика или обстоятельств.

Наконец, хороший процесс ведет к успеху. Если выбранный процесс ограничивает возможности достижения успеха, его необходимо пересмотреть.

Марк Сандерс, изобретатель складного велосипеда Strida, описывает на YouTube процесс, который он использовал при создании своего «детища»\*. Когда я смотрел это описание, я отметил следующее:

- Сандерс не думал о том, какие идеи хорошие, а какие плохие. Он создавал эскизы, чтобы исследовать все возможные идеи.
- Когда эскизов накапливалось много, он оценивал их, исходя из первоначальных целей создания продукта. Так отбирались лучшие идеи.
- Эскизы использовались только до этого этапа. Чтобы убедиться в работоспособности идей, ему приходилось создавать модели или прототипы.
- Эскизы сыграли решающую роль в усовершенствовании проекта и разработке модели Strida2.

В своем видео Марк описывает достаточно типичный процесс прототипирования, включающий следующие элементы:

- создание эскизов;
- оценку;
- создание модели;
- испытание.

В моей компании Messagefirst используется подход, подобный описанному Марком, но с небольшими отличиями. Мы берем

---

\* См.: <http://www.youtube.com/watch?v=j5tFFIBDzVs>.

## На что похож процесс прототипирования

страницу из книги дизайннерской студии и применяем модель с демонстрацией проекта и его критической оценкой. Так что наш видоизмененный процесс выглядит так:

- создание эскизов (набросков на демонстрационных досках или бумаге, программного кода);
- демонстрация и критическая оценка;
- моделирование (прототипирование);
- тестирование.

Одна из наших целей — быстрый цикл последовательных итераций и развития. Поэтому мы устанавливаем довольно короткую продолжительность первых двух стадий — создания эскизов; демонстрации и критической оценки. Это обеспечивает развитие процесса, повышает нашу производительность и эффективность прототипирования.

Наш процесс прототипирования также *итеративен* и *эволюционен*. Мы делаем набросок, показываем и критически оцениваем его, создаем прототип, показываем и критически оцениваем его, снова создаем прототип, а затем проводим тесты (рис. 2.1). Затем мы повторяем эту цепочку действий снова.

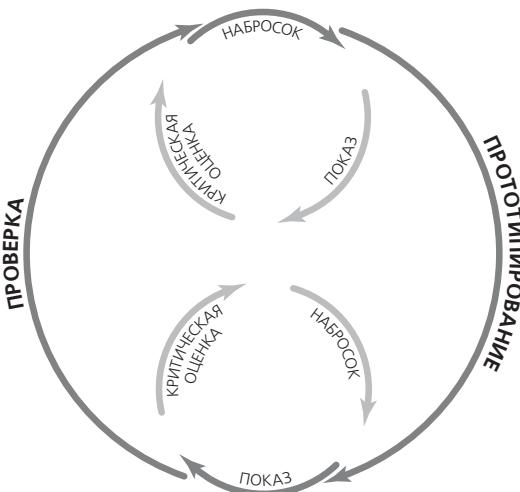


РИС. 2.1.  
Диаграмма  
итеративного процесса  
разработки  
и критических оценок

## На что похож процесс прототипирования

Вероятно, вы обратили внимание на циклический характер и неоднократное повторение двух шагов: создания наброска; демонстрации и критики. Наброски создаются и обсуждаются на протяжении всего процесса прототипирования. Фактически, когда мы показываем и оцениваем наши прототипы (как в компании, так и с привлечением клиента), мы делаем набросок наших исправлений.

Теперь давайте подробно рассмотрим наш процесс прототипирования, начиная с создания наброска.

### Часть 1: создание наброска

Создание наброска — производительная часть прототипирования, и ваша цель на этой стадии — «вытащить» идеи из головы и материализовать их.

Я предпочитаю устанавливать рамки для «времени делать наброски». Это заставляет сотрудников работать быстро и не вдаваться в детали.

Цель создания набросков — не конкретизация идей (этим мы занимаемся на стадии разработки прототипа), а определение концепций, «вытаскивание» их из головы максимально быстро и переход к следующей стадии.

#### **СОВЕТ** КОЛИЧЕСТВО ВАЖНЕЕ КАЧЕСТВА

При создании наброска не старайтесь разделить идеи на хорошие и плохие. В данный момент ваша цель — изучить идеи. На стадии создания набросков количество важнее качества. Качество придет позже.

Наброски обычно делаются начерно, иногда они неполны и откровенно схематичны, как видно из рис. 2.2. Не старайтесь сделать все без изъяна. Попытайтесь материализовать ваши идеи.

#### **СОВЕТ** БЫСТРЫЙ И ЯРОСТНЫЙ

Установите ограничения времени на создание набросков.

Я предпочитаю отводить на это 10–30 минут, а затем переходить к показу и критической оценке. Такой короткий период заставляет фокусироваться на продуктивных идеях, не вдаваясь в подробности.

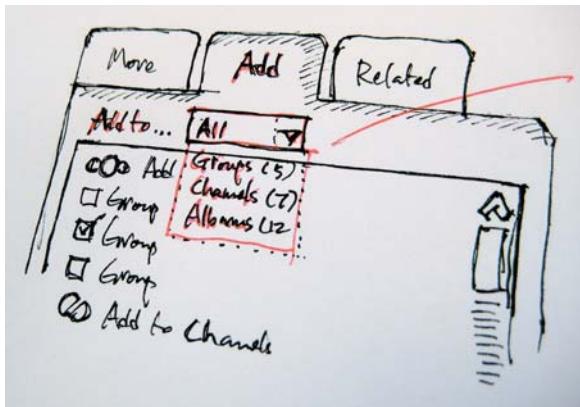


РИС. 2.2.  
Набросок модуля настроек  
для видео на Vimeo,  
демонстрирующий толщину  
линий и пометки\*

Большинство наших набросков делается на бумаге с использованием специальных планшетов. Планшет — просто лист бумаги с набором небольших окошек-шаблонов. Он подобен планшету для раскадровки.

Вот как выглядит один из наших планшетов для набросков (рис. 2.3).



РИС. 2.3.  
Пример планшета  
с несколькими набросками

Главная разница между планшетом для раскадровки и планшетом для набросков в том, что первый используется для последовательного изложения некой истории, а второй, наоборот, содержит набор идей, а их последовательность не имеет значения.

\* [www.flickr.com/photos/soxiام/2532060829/](http://www.flickr.com/photos/soxiام/2532060829/).

## На что похож процесс прототипирования

### СОВЕТ ХОРОШАЯ ИДЕЯ НЕ ПРИХОДИТ ОДНА

Использование планшета для набросков — отличный способ выдать несколько идей. Небольшое пространство стимулирует обдумывание отдельных элементов интерфейса. Такой планшет также подойдет для небольшой раскладовки, если необходимо описать проект AJAX или RIA.

Большинство наших набросков делается на бумаге, но иногда мы выполняем их на демонстрационных досках или в программном коде. Выбирайте среду, которая вам удобна, — лишь бы был результат.

Рассмотрим преимущества и недостатки набросков на бумаге, лекционных досках и в программном коде.

### *Наброски в коде*

Наброски можно делать не только на бумаге или лекционных досках. Разработчики часто делают их в привычной им среде — программном коде (это тоже возможно).

Одно из преимуществ набросков в коде — возможность их превращения в прототип. В условиях роста числа JavaScript-библиотек, CSS-фреймворков и шаблонизаторов вроде Ruby on Rails создавать наброски в коде становится все легче.

### Преимущества

- Делать наброски в коде все легче, поскольку инструментов для этого становится больше.
- Наброски «оживают» — их можно опробовать на практике.
- Это дает возможность эффективно использовать написанный код.

### Недостатки

- Не каждый умеет писать код.
- Для этого необходим компьютер.
- Возможностей для совместной работы меньше, чем при использовании бумаги или лекционной доски.

- Это занимает больше времени, чем создание набросков на бумаге или лекционной доске.

### *Создание набросков на лекционной доске*

Одно из главных преимуществ создания набросков на лекционной доске — возможность совместной работы. Кто угодно может принять участие в обсуждении и предложить собственный вариант.

#### **Преимущества**

- Обеспечивает возможность совместной работы.
- На лекционной доске может рисовать каждый.
- Одновременно могут участвовать несколько человек.
- Не нужен компьютер.
- Внести изменения очень просто: достаточно стереть рисунок и изобразить что-то новое.

#### **Недостатки**

- Перенести лекционную доску сложнее, чем код или лист бумаги.
- Наброски статичны.
- Перенести наброски с доски на рабочие материалы иногда сложно\*.

### *Создание набросков на бумаге*

Создание набросков на бумаге остается моим излюбленным методом. Это обеспечивает примерно такие же возможности для совместной работы, как и лекционные доски. Однако листы бумаги очень легко перемещать и работать на них можно где угодно.

#### **Преимущества**

- Обеспечивает возможность совместной работы.
- На бумаге может рисовать каждый.

---

\* Эскизы можно зафиксировать с помощью фотоаппарата или интерактивных досок Smartboard.

## На что похож процесс прототипирования

- Может участвовать несколько человек одновременно.
- Не нужен компьютер.
- Внести изменения просто — достаточно добавить в рисунок нужные элементы или взять другой лист бумаги.
- Этим можно заниматься когда угодно и где угодно.
- Наброски легко перемещать.

## Недостатки

- Наброски статичны.

## Часть 2: демонстрация и критика

Демонстрация и критика — вероятно, наиболее важная часть процесса прототипирования. На этой стадии мы сосредоточиваемся на качестве.

Метод демонстрации и критики я освоил в студенческие годы, когда изучал графическое проектирование. И хотя потом я выбрал специальность «английский язык и когнитивная психология», я никогда не забывал ценные уроки демонстрации и критики.

Цель на этой стадии — найти лучшие идеи. Вы показываете сильные стороны своей концепции, а ваши коллеги обращают внимание на области, которые требуют большей ясности или доработки. В этом и заключается суть стадии: обсудить, оценить и двигаться дальше.

Представляя наброски на обсуждение, мы часто крепим их к стене, как показано на рис. 2.4.

### *Рекомендации по демонстрации и критике*

**Будьте кратки.** Как я упоминал ранее, демонстрация и критика — вторая стадия, для которой мы устанавливаем жесткие рамки. Фактически на нее выделяется еще меньше времени, чем на набросок. Я стараюсь ограничивать время демонстрации двумя минутами, а обсуждение — тремя.



**РИС. 2.4.**  
**Показ набросков в ходе их критического разбора в дизайнерской студии**

**Не тратьте слишком много времени.** Я знаю, что это звучит отчасти парадоксально. Но цель здесь — быстро получить результат и двигаться дальше. Вы сможете детализировать свои идеи позже.

**Три минуты на демонстрацию идеи.** Лимит времени в три минуты на демонстрацию идеи означает, что вы должны будете сосредоточиться на сильных сторонах. А если вы не можете объяснить суть идеи за это время, вероятно, что-то с вашей идеей не так.

**Две минуты на критику.** Ваши коллеги получают две минуты на критику вашей идеи. В это время они должны показать две-три сильные и одну-две слабые ее стороны.

**Делайте заметки.** Пишите прямо на набросках прототипа. Используйте критические замечания ваших коллег, чтобы уточнить и усилить вашу концепцию.

### Часть 3: прототип

К началу этой стадии процесса вы набросали свои идеи, показали и обсудили их, отобрали только самые сильные варианты. Их прототипы вы и будете создавать.

Именно на стадии прототипирования вы начинаете детализировать свои проекты и выяснить, какие из них действительно жизнеспособны.

## На что похож процесс прототипирования

Когда я создаю прототип, то рассматриваю следующие вопросы:

- использую ли я инструменты или среду, с которыми мне удобно работать;
- есть ли у меня возможность эффективно передать *аудитории* этого продукта или его *потребителям* то, что я считаю нужным;
- каким временем я располагаю;
- какая степень точности необходима.

На деле способ прототипирования не важен. Большинство моих прототипов созданы с помощью HTML и AJAX, но это объясняется характером моей работы и потребностями клиентов. Я делал прототипы с помощью Flash, Keynote и на бумаге.

Когда прототип готов, я повторяю показ и критическое обсуждение. Я использую ту же базовую модель, показывая каждый раз только часть прототипа и предлагая клиенту или пользователю оценить ее. Основная разница в том, что я отвожу больше времени на показ и критику. В остальном процесс не меняется.

### СОВЕТ ПРОЕКТ И НАБРОСОК

Проектируйте свой прототип на лекционной доске в ходе показа и обсуждения наброска. Тогда вы сможете по ходу дела вносить изменения в прототип.

## Часть 4: тестирование

Я деляю тестирование на две категории: тестирование с клиентами и тестирование с конечными пользователями.

### Тестирование с клиентами

Тестирование с клиентами сочетает в себе показ, критическое обсуждение и создание набросков. Встречи делятся обычно 1,5–3 часа в зависимости от сложности прототипа.

В ходе встреч я, как обычно, показываю по одному элементу прототипа и обсуждаю его. Вместо списка я использую планшет, чтобы сделать

замечания к наброскам. Замечания в большинстве своем представляют собой небольшие метки или краткие комментарии от руки.

Фиксация изменений на набросках гарантирует, что мы не будем использовать для этого таблицу. Текстовые замечания легко понять неправильно. Наброски помогают снизить этот риск. Их создание допускает сотрудничество, и клиент может внести свой вклад.

Обычно я следую отработанной схеме при определении необходимых изменений в ходе процесса: больше набросков, меньше записей.

Когда этап рассмотрения проекта закончен, клиент получает доступ к прототипу. Я прошу потребителя опробовать прототип в течение следующих двух или трех дней.

Одна из целей создания прототипа — дать целевой аудитории возможность испытать его. Я хочу, чтобы клиент получил опыт работы с прототипом, использовал его. В ходе использования он либо обнаружит новые спорные моменты, либо поймет, что аспекты, казавшиеся ему спорными, на деле к таким не относятся.

### *Тестирование с конечными пользователями*

Тестирование с конечными пользователями — стандартный тест на удобство использования: 8–12 участников, 5–6 сценариев, аудио- и видеозапись, анализ, а после тестирования — сообщение о результатах. В главе 12 подробнее рассказано о тестировании прототипов.

При любом тестировании я учитываю полученную в его ходе информацию в следующей версии прототипа.

### Резюме

Я рассказал о процессе прототипирования в Messagefirst. Надеюсь, это поможет новичкам в данной области увидеть, как прототипирование улучшает процесс проектирования. А бывалые профессионалы, возможно, найдут несколько полезных советов для усовершенствования используемых ими процессов прототипирования.

Прототипирование — процесс, а не продукт, средство достижения цели. Вот несколько важных моментов, о которых нельзя забывать:

- создание набросков — ключевая часть процесса прототипирования;
- используйте метод дизайнерской студии: составление набросков; показ и критическое обсуждение. Это помогает быстро и последовательно улучшать прототип;
- начинайте с количества, исследуя множество идей. Качество придет позже.

В следующих двух главах я рассмотрю 5 различных способов прототипирования и 8 руководящих принципов его выполнения.

## ГЛАВА 3



# Пять моделей прототипирования

Модель № 1: коммуникация	52
Модель № 2: доработка и проектирование	57
Модель № 3: «продайте» свою идею в компании	58
Модель № 4: тестирование удобства использования	60
Модель № 5: оценка технической осуществимости и стоимости	62
Резюме	67

**E**ще до начала работы над книгой я знал, что должно быть несколько разных видов прототипирования. Это подтвердит любой человек с приличным опытом в нашей области. И хотя я не могу утверждать, что открыл новый вид прототипирования или новое применение этому методу, обсуждение его с другими практиками привело к тому, что я стал воспринимать его по-новому.

В моих частых разговорах с другими практиками и поставщиками то и дело всплывали обычные сомнения по поводу функций прототипов: они могут передавать другим идеи или концепции, привлекать внимание к идее в компании, помочь продать вашу идею пользователям или выяснить, что именно можно реализовать.

А когда мы с Джедом Будом обсуждали взаимодействие с пользователями, возникло гениальное определение. Джед помог мне понять, что такое прототипирование: *способ разработки своих проектов*.

Как я уже говорил, это вовсе не было внезапным открытием нового способа использования метода. На самом деле в ходе разговора я постепенно осознал, что уже давно применяю такой подход. Я никогда не задумывался об этом до разговора с Джедом. Для меня как практика данное определение, пожалуй, самый веский довод в пользу прототипирования.

## Модель № 1: коммуникация

Как я уже говорил, наличие типовой платформы для обсуждения снижает вероятность неправильного толкования. Прототипы — типовой язык образов. В любом конкретном проекте есть несколько разных групп, объединенных общими требованиями: выпускать продукт, поддерживать жизнеспособность компании и зарабатывать деньги. Но не забывайте, что команды говорят на разных языках и имеют собственные планы.

Доводилось ли вам бывать в одной комнате с группой инженеров и маркетологов? Они не могут общаться друг с другом. То есть говорить то они могут, но беседы не получается. Они друг друга не понимают.

Они используют разные языки. Пригласите в одну комнату разработчика интерфейсов, архитектора информационных систем и специалиста по удобству использования — и можете запастись попкорном и колой, усесться поудобнее и наслаждаться представлением.

Что же делать? Приготовьте бумагу или лекционную доску и проведите короткое занятие по совместной разработке прототипов. Оно не обязательно будет блестящим, но люди должны начать взаимодействовать и понять суть проекта. Пусть они работают вместе, делают наброски.

Прототипирование как основа сотрудничества имеет ряд преимуществ. Во-первых, оно создает канал общения между двумя группами. Во-вторых, оно предоставляет им возможность научиться общаться друг с другом. Но главное — вовлечение проектировщиков и разработчиков в совместную реализацию идей помогает им *строить взаимоотношения* друг с другом.

Когда проектировщики и разработчики связаны общими отношениями, результат оказывается отличным. Первые скорее дадут вторым попробовать что-то новое, чтобы расширить границы, использовать имеющиеся технологии таким образом, чтобы можно было создать нечто неожиданное, обойти имеющиеся ограничения или изобрести новую технологию (необходимость — мать изобретений). Проектировщики помогут разработчикам понять, что они действительно *сделать могут*, а чего *не могут*.

Разработчики помогут понять сущность технологий, дав проектировщикам доступ к возможностям, о которых те даже не подозревали. Но они могут также «держать проектировщиков в узде»: не позволять им заплыть слишком далеко в море идей и погибнуть во время бури. Разработчики помогают проектировщикам увидеть, какое решение разумно и реализуемо.

### СОВЕТ ИСПОЛЬЗУЙТЕ ВОЗМОЖНОСТЬ СОТРУДНИЧЕСТВА

Прототипирование в сотрудничестве с коллегами — отличное упражнение для создания команды. Опробуйте этот подход. Посадите в одной комнате несколько проектировщиков

## Модель № 1: коммуникация

и разработчиков и поручите им создание прототипа как упражнение на выстраивание отношений. Такая платформа для общения даст немедленную выгоду. Скрытая выгода в том, что сотрудничество поможет проектировщикам понять, какое решение разумно, а разработчики увидят, что они действительно умеют делать, а что нет.

Другая важная выгода от прототипирования как платформы для общения проявляется, когда команды разработчиков находятся далеко друг от друга. По мере роста компании команда становится сегментированной. Группа маркетинга сидит на одном этаже, проектировщики — на другом, разработчики — на третьем. Позже сотрудники могут оказаться в разных зданиях, а то и в разных городах. К тому же существует офшорный бизнес — ваши разработчики могут оказаться в Индии, России или Китае.

В таких ситуациях общение между сотрудниками затруднительно. В современном мире оно происходит преимущественно с использованием телеконференций, служб мгновенных сообщений или электронной почты, даже если вы сидите в одном здании. А если сама компания находится в Калифорнии, а группа разработчиков — в Восточной Европе или в Азии, то возникают проблемы разных часовых поясов и языкового барьера. Они могут быть решены путем создания прототипов, помогающих установить контакт.

Прототипы помогают устраниить языковой барьер, действуя как платформа для коммуникаций, *показывая*, а не *говоря*. Прототипы также не зависят от часовых поясов. Можно потратить больше времени на *показ* ваших требований команде разработчиков и меньше на *рассказ*. Обратное тоже верно: команда разработчиков может создать прототип и показать вам, как они поняли ваши пожелания.

Такой метод «покажи и расскажи» Роберт Хекман-младший называет протокастами (своего рода аналог подкастов\*). Хекман создает серию экранов, обычно в программе OmniGraffle, и затем записывает симуляцию работы важных элементов прототипа.

---

\* Подкаст — аудио- или видеофайл в стиле радио-/телепередачи, распространяемый в интернете. Прим. ред.

## Протокоммуникации

*Роберт Хекман-младший* ([www.rhjr.net](http://www.rhjr.net), [www.miskeeto.com](http://www.miskeeto.com))

Несомненно, прототип может кратко рассказать о проекте взаимодействия или даже о приложении в целом. Но создание прототипов — не только материализация идеи для показа заинтересованным сторонам, как будет работать приложение, или оценки реализуемости проекта.

В конечном счете прототип — средство общения. Его можно использовать для маркетинга или закрытия пробелов в общении между несколькими группами, работающими над проектом, а также для показа функций отдельных мелких элементов, которые нельзя описать словами или статичными картинками. В таких случаях при использовании правильных инструментов можно создавать прототипы, способные стать неотъемлемой частью проектирования.

Рассмотрим конкретный пример.

При работе над многофункциональными интерфейсами, например RIA или интерфейс «по требованию» в стиле Web 2.0\*, иногда сложно детально описать многочисленные состояния одного экрана или одиночное взаимодействие с экраном через отдельные изображения. Чтобы разделить эти взаимодействия на удобоваримые компоненты, я обычно создаю раскадровку — серию эскизов, показывающих состояния экрана по мере взаимодействия. И, конечно, я стараюсь как можно лучше документировать взаимодействия, подробно отражая варианты использования в «документе описания проекта» (Design Description Document, см. [www.rhjr.net/ddd](http://www.rhjr.net/ddd)). Однако часто этого недостаточно.

При использовании раскадровок возникают две проблемы. Во-первых, мне необходимо визуализировать переход из одного состояния в другое,

\* Web 2.0 — сетевые проекты, информацию для которых подбирают и уточняют непосредственно пользователи. Прим. ред.

### Протокоммуникации (продолжение)

но я не имею возможности «проделать» этот путь воочию. Во-вторых, то же приходится делать клиентам. И хотя во многих случаях раска-дровки приносят отличный результат, прототипы «оживляют» идею и дают возможность каждому изучать проект таким способом, кото-рый не может быть реализован с использованием только статичных изображений, даже в формате раскадровки.

Теперь я иногда использую для работы по проектированию взаимо-действия программу OmniGraffle. Она поддерживает назначение некоторых базовых функций на клики по каркасным представлениям и диаграммам. Это позволяет быстро создать и передать другим очень схематичный набросок прототипа, который можно «прощелкать», в виде документа PDF в любой момент. Таким образом, я не только могу представить документ, который каждый в состоянии открыть и опробовать. Создаваемый мною прототип выше уровнем, чем обыч-ный видеофильм, показывающий работу прототипа и записанный с помощью инструментов SnapzProX (Mac) или Camtasia (Windows).

Эти «протокасты» ([www.rhjr.net/shorty/protocasting](http://www.rhjr.net/shorty/protocasting)), как я их на-зываю, — просто запись моего взаимодействия с прототипом, со-проводжающаяся голосовыми комментариями по поводу выполняемых действий. Это могут быть размышления, технические ограничения, предложения по улучшению и т. д.

Протокасты — отличный инструмент не только для более четкого объяснения взаимодействия, чем при использовании раскадровки, но и для обзора удобства использования и важных исследований других характеристик. Я просто фиксирую изображение экрана, когда завер-шаю одну из задач приложения, и надиктовываю свои критические замечания, прежде чем переходить к следующей. Я даже могу гово-рить от лица гипотетического пользователя, например: «На самом деле я не понимаю, что означает этот значок... Может, я лучше нажму

### Протокоммуникации (окончание)

на эту кнопку и посмотрю, что произойдет... Ой, это не то, чего я хотел».

Для создания прототипа в формате PDF с помощью OmniGraffle и записи протокола требуется очень мало усилий и времени. Я могу делать это часто в рамках рабочего процесса. А время, требующееся на создание прототипов и протоколов, с лихвой компенсируется экономией минут и часов на обсуждениях, которые неизменно давали только схематичные результаты.

Вместо того чтобы тратить много времени на сложный, многофункциональный прототип полного приложения, я создавал прототипы и протоколы только для тех элементов интерфейса, которые действительно требовали пояснений. Эти составляющие обычно очень малы по объему, поэтому на их создание требуется немного времени; но при этом используются все преимущества прототипирования.

В целом это дешевое, быстрое и безболезненное решение помогало мне обмениваться идеями и критическими замечаниями в случаях, когда слов недостаточно, а полнофункциональный прототип избыточен.

Эффективное общение — важный элемент успешного проекта. Прототипы и протоколы помогают всем следить за ходом разработки и не отставать от коллег при небольших затратах времени и усилий.

## Модель № 2: доработка и проектирование

В перепроектировании используются две модели: «легкая ретушь» и радикальная переработка.

В первом случае сохраняется облик проекта, вносятся небольшие обновления. Может быть, вводится несколько новых функций,

### **Модель № 3: «продайте» свою идею в компании**

улучшений или исправлений, но в целом проект остается прежним. Это больше похоже на «причесывание», чем на перепроектирование.

Радикальная переработка — совсем иной случай. Вы переходите устоявшиеся границы, пробуете радикально новые идеи, новые концепции проектирования. Вы получаете новое лицо, сбрасываете несколько килограммов, выбираете новый гардероб — даже друзья не всегда узнают вас. Результат может быть как отличным, так и ужасным.

Прежде чем начать радикальную переработку, ее надо протестировать. Вам необходим способ исследования различных вариантов проекта, определения методов, тестирования и точного определения характеристик. Использование прототипов — отличный способ доработать проекты, протестировать их, определить, какой из них будет успешным, и конкретизировать его детали.

### **Модель № 3: «продайте» свою идею в компании**

В большинстве случаев работа на благо потребителей и обеспечение большего удобства использования в итоге оказываются крайне выгодными. Чтобы проверить эту гипотезу на практике, консультационная фирма по вопросам удобства использования Teehan+Lax создала фонд UX Fund, который вкладывает средства в компании, нацеленные на обеспечение максимального удобства использования их продуктов, чтобы посмотреть, каковы будут результаты в сравнении с другими компаниями в отрасли.

В ноябре 2007 года этот фонд показал годовые темпы роста в 39,5%\*. Эта величина сама по себе впечатляет. Она будет еще более впечатляющей, если сравнить ее с общим состоянием рынка в то время. В конечном итоге вложения в удобство использования наиболее интересны для бизнеса, но не все в это верят.

Несколько лет назад я участвовал в разработке одного из первых коммерческих продуктов VoIP (голосовая связь в интернете) на рынке.

---

\* [www.teehanlax.com/blog/?p=293](http://www.teehanlax.com/blog/?p=293).

На самом деле это была моя вторая попытка в данной области; первая стала жертвой краха доткомов.

При поиске решения чаще всего рассматриваются конкурирующие варианты. Другой метод — посмотреть на похожие продукты (например, мобильные телефоны, домашние телефоны, телефоны для офиса). Еще один метод, мой любимый, — исследование новых моделей взаимодействия и решений. Каждая проблема имеет много вариантов решений, и выбрать лучшее иногда сложно.

Рассмотрим одну из возможностей этого продукта VoIP, имеющуюся почти в любом телефоне, — быстрый набор номера. Обычные телефоны дают возможность вводить номера в список быстрого набора только по одному. В значительной степени это обусловлено малой площадью экрана телефона (это относится и к стационарным, и к радиотелефонам, и к мобильным телефонам). Рабочая площадь экрана компьютера куда больше, и это увеличивает число возможных вариантов.

Существуют традиционные модели ввода в веб-приложения нескольких номеров или фрагментов данных. Главное — разместить на экране несколько десятков полей для ввода. Это помогает достичь цели, но десятки полей для ввода номеров нельзя назвать простым и аккуратным решением. Поэтому у меня появилась другая идея: добавлять поля по мере необходимости. Начните с простого дизайна и модифицируйте его, когда это потребуется.

Я описал свою идею менеджеру продукта и команде разработчиков, а затем показал им набросок (рис. 3.1). Рассмотрев его, они поняли концепцию. Менеджменту идея понравилась, но команда разработчиков хотела сделать группу статичных полей, поскольку это проще. (Ниже я расскажу, что произошло дальше.)

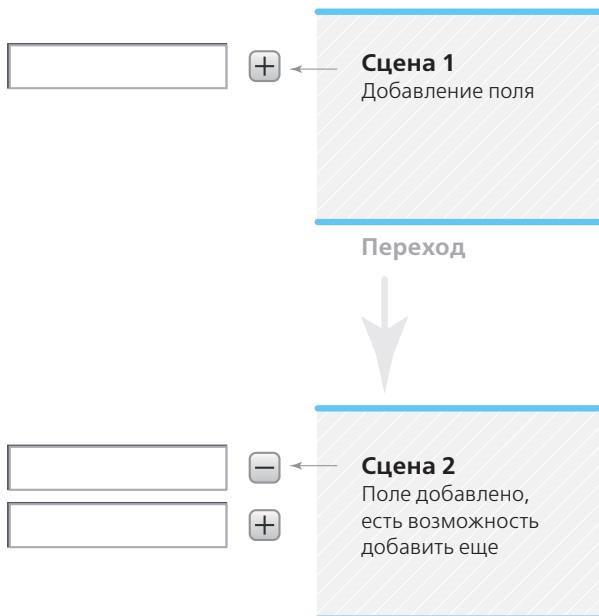
Прототипы — инструмент «продажи» технической реализуемости и ценности вашей концепции. В некоторых случаях они могут оказаться настолько эффективными, что высшее руководство даст распоряжение команде разработчиков просто создать их.

## Модель № 4: тестирование удобства использования

### СОВЕТ ЛУЧШЕ ПОКАЗАТЬ, ЧЕМ РАССКАЗАТЬ

Не просто рассказывайте об идеях, а показывайте различные варианты проекта. Сделайте схематичный прототип разных вариантов возможных решений. Предъявите его другим членам команды и получите от них критические отзывы. Так можно использовать прототип в качестве инструмента для создания проектировочных решений.

РИС. 3.1.  
Набросок  
динамически  
добавляемых полей



## Модель № 4: тестирование удобства использования

Помните программу VoIP, о которой шла речь выше? Нам было нужно, чтобы группа разработчиков «купила» ее. Они не были уверены в пользе предложенного решения, и мы решили провести тесты на удобство пользования. Это помогло бы оценить, хороша концепция или нет. Если она эффективна — отлично. Мы смогли бы представить полученные данные руководству и разработчикам как сильный аргумент в пользу своей идеи. Если нет, то мы бы об этом быстро узнали и смогли бы рассмотреть другой вариант.

## Модель № 4: тестирование удобства использования

До начала тестов я был уверен, что люди получат рабочую модель с динамическим вводом, но не знал, будет ли она понятна. Я ожидал, что кто-то примет ее сразу, а кто-то поймет только в ходе использования.

Каркасное представление не вызывает бурных восторгов. Я никогда с таким не сталкивался.

Дэвид Верба, *Adaptive Path*

Мы создали набор бумажных прототипов и протестируем его с восьмью участниками. Каждому мы давали набор задач на добавление полей с использованием традиционной модели, а затем предлагали воспользоваться новой, дающей возможность добавлять поля по мере необходимости.

Половине участников сначала показывали традиционную модель, а затем новую. Остальным — в обратном порядке. Мы хотели уменьшить предвзятость и гарантировать, что тестируются именно множественные действия, например добавление нового контакта, дополнительных номеров телефонов и адресов электронной почты в контакт, добавление телефонных номеров в список быстрого набора.

Нас приятно удивило, что все 12 участников сразу разобрались в модели взаимодействия по требованию. Каждый вводил информацию в текстовое поле, выбирал действие «добавить» и смотрел, как мы моделировали появление нового поля.

Я считаю, что успех объясняется рядом факторов. Во-первых, на экране был только один вариант действия, если не считать кнопки «Сохранить»: кнопка со значком «плюс», которую все воспринимали как кнопку добавления. (В данном случае мы также использовали всплывающую подсказку.) Уменьшение количества вариантов возможных действий сокращает вероятность непонимания. Во-вторых, размещение этого значка рядом с полем ввода помогало сделать вывод: «Это действие относится к этому полю».

## Модель № 5: оценка технической осуществимости и стоимости

Мы могли бы потратить часы, ходя кругами вокруг команды разработчиков и пытаясь «продать» им наше решение. А на деле им надо увидеть пример и результаты, а не просто услышать о них. Тогда они станут более восприимчивы, хотя сомнения останутся.

Однако если тест на удобство использования покажет положительный результат, им будет сложно заявить, что модель нерабочая, и в конце концов они «купят» идею.

Прототипирование дает нам возможность донести свою концепцию до других и принять обоснованное решение.

## Модель № 5: оценка технической осуществимости и стоимости

Итак, у вас есть действительно хорошая концепция. Вы поработали с визуальным дизайнером (или вы сами дизайнер) и получили блестящий прототип. Но остается главный вопрос: можно ли это реализовать?

Множество отличных проектов было отвергнуто разработчиками или технологическими командами, убежденными, что реализовать эти идеи не удастся. Если вы проектировщик и у вас есть отличный проект без прототипа, вам предстоит жестокая и упорная битва.

Иногда на вашей стороне высшее руководство компании, которое заставит разработчиков реализовать ваш проект. А если нет? Если проект действительно сложен? Если в нем непросто разобраться? Тогда необходимо задуматься о создании прототипа.

Чтобы показать техническую реализуемость, не требуется уровень качества, близкий к реально работающей версии. Во многих случаях, если не почти всегда, можно имитировать или симулировать значительную часть пользовательского интерфейса. Этого достаточно, чтобы показать концепцию менеджменту или команде разработчиков, дать им возможность «купить» ее и определить, возможно ли ее реализовать и стоит ли вкладывать в нее деньги.

## Модель № 5: оценка технической осуществимости и стоимости

Реализуемость и целесообразность одинаково важны. Имея программу, вы можете сделать все, что захотите. Нужно лишь вложить в нее достаточно времени и ресурсов. Но в этом и кроется ловушка. Для бизнеса в целом может быть значительно хуже, если на разработку некоторой возможности или нескольких возможностей будут потрачены время и усилия. Окупаемость инвестиций — весомый аргумент и для разработчиков, и для руководства.

Для участников технологической команды окупаемость инвестиций означает соотношение размера кода и его функциональности. Работоспособность они определяют на основе технических аспектов. Один мой подчиненный, инженер, как-то заявил, что определяет работоспособность так: если он открывает окно терминала, то должен увидеть, что на компьютере запущен сервер Apache. А я ожидаю увидеть страницу в окне моего браузера. И обе эти ситуации означают, что Apache «работает».

Для бизнеса окупаемость инвестиций — соотношение вложений и прибыли. Затраты могут быть фиксированными, например при покупке компьютерного оборудования и программного обеспечения для создания рабочего места, или плавающими, например стоимость обслуживания или привлечения и удержания клиентов. Чем сильнее мы хотим получить чистое соотношение, тем больше практического опыта нам нужно, чтобы достичь «просветления»; на деле же бизнес несет финансовую ответственность перед «просветленными» работниками.

Большинство наших прототипов имели качество конечного продукта. HTML, CSS и JavaScript часто используются командой разработчиков для создания финальной версии. Нельзя сказать, что это типично, зато реализуемо.

Если вы создаете прототип с качеством конечного продукта, вопросов о реализуемости не возникает: разработчики получают презентационный образец с полной функциональностью. Единственное препятствие в этом случае — необходимость убедить руководство, что наша

## Модель № 5: оценка технической осуществимости и стоимости

концепция хороша и время, необходимое для ее реализации разработчиками, ценно для компании. И эту проблему несложно решить.

Когда мы отдаем прототип пользователям, они видят немедленную выгоду. Предоставьте им несколько дней на тестирование — и они ваши. Если пользователи «купят» прототип, руководство последует их примеру.

### Практический пример: IntraLinks

*Тодд Заки Варфел*

Компания IntraLinks захотела переработать свое веб-приложение для виртуальных защищенных комнат данных. Приложение практически не менялось в течение пяти лет, его нужно было обновить. Какие-то необходимые изменения всегда очевидны; но требовалась помочь извне, чтобы вскрыть не столь явные проблемы. Поэтому они обратились в мою компанию Messagefirst. После четырех дней тестирования мы составили обширный список рабочих и нерабочих функций и представили ряд рекомендаций. Вооружившись этой информацией, команда проектировщиков IntraLinks приступила к задаче. За полтора года они усовершенствовали приложение и создали работающий, качественный, многофункциональный прототип.

Это был один из лучших прототипов на моей памяти. Команда проектировщиков IntraLinks выполнила радикальный переход от веб-приложения, которое вело себя как портал, к веб-приложению, похожему на обычное приложение.

Итоговый вариант стал результатом эволюции прототипов. Сотрудники IntraLinks не тратили полтора года на один-единственный прототип. Они показали мне ряд предыдущих версий. Многие из тех прототипов значительно отличались от последней версии, которую они были готовы представить пользователям. Они подвергали каждый прототип тестам, и результаты получались разными.

### Практический пример: Intralinks (продолжение)

В некоторых случаях другие проектировщики не чувствовали, что прототип понятен. Иногда руководству не нравилось направление, задаваемое прототипом будущему продукту. Поэтому был создан ряд прототипов, и постепенно разработчики пришли к тому варианту, который они хотели показать пользователям и услышать их отзывы. Это эволюционный процесс.

Возможно, вы удивитесь: «Если это действительно так здорово звучит, то зачем они стали этот прототип тестировать?» Потому что для них это была новая концепция проектирования. Точнее, имелось несколько радикально новых концепций. Проектировщики хотели увидеть реакцию пользователей. Некоторые потребители применяли предыдущую версию в течение пяти лет. Даже если новый проект оказался «лучше», его запуск стал бы катастрофой, если бы представители целевой аудитории не смогли с ним работать.

Во-первых, старая версия была ориентирована на страницы: страница обновлялась после каждого действия, сопровождаемого нажатием кнопки. Это не значило, что приложение нерабочее; просто оно устарело и стало неэффективным. А в новой версии использовалась экранная модель с частично отирующими фиксированными состояниями и богатыми возможностями взаимодействия, которую часто называют RIA. В ней была масса современных функций, например перетаскивание, использование правого клика мышью, контекстные меню и автообновление. Как и в любом случае кардинального изменения интерфейса, разработчики хотели узнать, не испортили ли они продукт. А если приложение улучшилось, то насколько?

Во-вторых, у них были актуальные вопросы. Смогут ли пользователи воспринять новую концепцию? Обнаружат ли они возможность перетаскивания? Как насчет кликов правой кнопкой и контекстных меню? Если они станут использовать эти возможности на одних участках

### Практический пример: IntraLinks (окончание)

приложения, на каких еще участках такие возможности будут уместны? Какие концепции не работают вовсе или нуждаются в улучшении?

При рассмотрении этих изменений и имеющихся вопросов функциональный прототип с высокой точностью приближения был лучшим способом протестировать проект приложения перед запуском продукта.

В целом участники позитивно отреагировали на новый дизайн.

Некоторые отметили, что он напоминает другие приложения. Многие возможности взаимодействия, отличавшиеся от таковых в предыдущей версии IntraLinks, были знакомы из-за схожести с проводником Windows, Outlook и другими программами. Команда проектировщиков IntraLinks выгодно использовала знания, полученные при работе с другими приложениями, в своем новом проекте.

Примерно половина участников обнаружила новые возможности: перетаскивание, правый клик и контекстные меню. Похоже, пользователи ожидали увидеть их в таком безупречном приложении. Те, кто таких функций не нашел, просто не думали о том, что они могут существовать в подобном веб-приложении. Однако когда их показали, участники очень обрадовались и стали искать их во всех частях приложения.

Прототип и тестирование принесли огромный успех, подтвердивший многие решения, которые были приняты при проектировании. Разработчики доказали, что можно успешно совершить переход от программы, похожей на портал, к приложению, и обосновали ценность взгляда изнутри на дополнительные новые концепции систем передачи сообщений и множественных состояний.

Для компании IntraLinks способность применять прототипы для изучения различных решений в ходе проектирования была первостепенной. Разработчики использовали прототипы для изучения решений, что особенно ценно при создании новой концепции дизайна или RIA.

## Резюме

Прототипы могут отличаться по качеству и функциональности, но наиболее типичное их использование включает:

- создание среды для совместного общения;
- переработку методом проектирования;
- «продажу» идеи своему начальнику или коллегам в компании;
- тестирование удобства использования;
- оценку технической реализуемости и стоимости.

Возможно, ваши прототипы используются сразу несколькими из этих способов. Понять, что прототип может быть платформой для общения, средством для тестирования удобства использования и оценки технической осуществимости и стоимости, несложно.

В следующей главе я опишу восемь руководящих принципов прототипирования.