



Ценность прототипирования

Прототипирование продуктивно	25
Прототипирование — возможность показать, рассказать, получить опыт	25
Прототипирование снижает вероятность неправильного восприятия	28
Прототипирование позволяет сэкономить время, усилия и деньги	29
Прототипирование сокращает объем напрасного труда	29
Прототипирование показывает реальные ценности	32
Резюме	35

Миллионы людей могут мельком заглянуть в будущее. Вспомните концепт-кары. Изготовители тратят годы и миллионы долларов на эти своеобразные творения. Большинство из них не попадает в серийное производство. Те, которые все же выпускаются, часто мало чем напоминают оригинал.

Автомобильная промышленность высококонкурентна. Инновации здесь — не только движение вперед, но часто одно из обязательных условий выживания. Каждый концепт — упражнение в дизайне, исследованиях. Разработчики выясняют, что можно сделать и что можно будет продать. Так и создаются *прототипы*.

Этот метод был важной составляющей автопромышленности многие годы. Концепт-кары обходятся дорого, но гораздо дороже переоборудовать все производственные линии и запустить модель, которая не найдет спроса на рынке. Риск слишком велик.

Сформулировать аргументы в пользу того, что прототипирование необходимо при разработке автомобиля или системы наведения ракет, несложно. Однако доказать, что создание прототипов программных средств даст выгоду при реализации проекта, гораздо труднее. Фактически это одна из самых сложных проблем.

В данной главе я опишу некоторые проблемы, которые возникают при попытке встроить прототипирование в существующие процессы проектирования и разработки. Я дам совет, как показать, что прототипирование поможет в раннем выявлении проблем, снижении рисков и экономии времени, усилий и денег.

Клиенты и менеджеры, не слишком хорошо знакомые с прототипированием, часто воспринимают его как затратное мероприятие, приносящее мало выгоды (или не приносящее ее вовсе). Чаще всего мне приходится отвечать на такой вопрос: «Как мне заставить босса или клиента заняться прототипированием? Они говорят, что у нас для этого нет ни времени, ни денег».

Если вы занимаетесь (помимо прочего) созданием сайтов, программных приложений или систем, в число компонентов которых входят

как оборудование, так и программное обеспечение, без прототипирования вам не обойтись. По мере увеличения сложности системы соотношение цены прототипирования и получаемой выгоды резко растет.

Прототипирование стоит денег. Но если вы его не используете, то упускаете возможности инноваций и существенной экономии. Выгоды перевешивают затраты.

Прототипирование продуктивно

Одна из основных ценностей прототипирования — его продуктивность. В процессе создания прототипа появляются сотни, если не тысячи идей. Некоторые из них блестящие, другие не очень. Я обнаружил, что даже идеи, на первый взгляд не кажущиеся блестящими, могут стать катализаторами отличных решений.

Процесс прототипирования часто приводит к инновациям и значительной экономии времени, усилий и денег. Он помогает материализовать идеи, создать нечто, что вы можете ощутить, испытать, проработать, протестировать (и просто поиграть с ним).

Прототипирование — возможность показать, рассказать, получить опыт

Если картинка лучше тысячи слов, то прототип лучше десяти тысяч. Он значительно расширяет возможности показа и рассказа, дает возможность испытать разработку.

Одно дело рассказывать и показывать раскадровки, и совсем другое — видеть в реальности.

Роберт Хокман-младший

Есть ряд способов документирования дизайна, включая описания требований, метод каркасного представления, визуальные компоненты и прототипы.

Стандартные модели документирования

Техническое задание. Обычно это документ, где объяснены технические и функциональные требования к системе.

В технических заданиях обычно много слов и мало иллюстраций. Они много *говорят*, но мало *показывают*. Недостаток визуальной симуляции часто приводит к неправильному пониманию задания. Чтобы снизить этот риск, в документ можно добавить изображения, но только статичные.

Каркасные представления. Вы видели архитектурные чертежи зданий? Это примерно то же, что и каркасное представление для программного обеспечения. Каркасные представления — изображение структуры функциональных страниц. Они показывают, какие элементы есть на странице и как они соотносятся друг с другом. Каркасные представления обычно выполняются в черно-белом варианте или в оттенках серого.

Каркасные представления в сочетании с подробным описанием поведения элементов лучше решают задачу, чем технические задания. Однако часто остаются пробелы, что приводит к пропуску деталей или неправильному толкованию.

Прототипы. Прототип — модель или симуляция, представляющая финальный вариант системы. В отличие от технических заданий и каркасных представлений, он дает возможность испытать разработку.

Некоторые технические ограничения, например на размер страницы в 100 КБ, могут быть незаметными в прототипе. Их можно зафиксировать во вспомогательном документе, гораздо меньшем по размерам, чем 60–200 страниц.

По сути, технические задания и каркасные представления недостаточны для того, чтобы показать сложную систему и рассказать о ней. Их можно использовать для простых систем, но в случае сложной системы возникнут серьезные трудности. Нередко эти два метода применяются вместе для создания «полной картины». Однако испытания в таком случае часто заканчиваются неудачей.

Сочетание аннотированных каркасных представлений с техническим заданием может представить исходное видение с точностью 70–80%. Остается слишком много возможностей для ошибок.

AJAX и Monkey Wrench

Что произойдет, если добавить AJAX* или другое приложение RIA (Rich Internet Applications — многофункциональные интернет-приложения)? Все начнет разваливаться на части, причем быстро. Ни техническое задание, ни аннотированные каркасные представления не могут отразить правила взаимодействий и переходов.

В основе традиционных взаимодействий лежат страницы. В AJAX и RIA часто используются состояния. Страница или экран могут содержать несколько слоев или виджетов, которые работают независимо друг от друга (и в то же время взаимосвязаны). Например, обновление канала RSS на странице не требует обновления всей страницы. Обновляется только виджет RSS, все остальные элементы остаются прежними.

Это дало многим разработчикам повод заявить, что парадигма страницы мертва, а новая парадигма — экран или состояние.

Переходы и анимация — еще одна проблема. Пытались ли вы описать самоисправляющийся переход в AJAX?*** Мое самое удачное описание, дополненное пассажами фокусника и взмахами воображаемой волшебной палочки, вызвало удивление и непонимание у публики.

AJAX и другие технологии RIA используются все чаще; потребность в прототипах и их ценность как средства взаимодействия при разработке существенно возрастают. Я бы сказал, что прототипы становятся крайне важны для достижения успеха.

* AJAX (от англ. Asynchronous JavaScript and XML, асинхронный JavaScript и XML) — использование «фонового» обмена данными браузера с веб-сервером в интерактивных пользовательских интерфейсах веб-приложений (позволяет ускорить и повысить удобство работы). Термин появился в 2005 году. *Прим. ред.*

** Подробнее см.: <http://developer.yahoo.com/ypatterns/richinteraction/transition/selfhealing.html>. *Прим. науч. ред.*

Прототипирование снижает вероятность неправильного восприятия

Возьмите 60-страничное техническое задание. Посадите в комнате 15 человек. Раздайте им документ. Пусть они его прочтут. Спросите у них, что именно вы разрабатываете. Скорее всего, вы получите 15 разных ответов. Теперь представьте, что в документе 200 страниц. Результат будет еще хуже.

Прототипы — гораздо более конкретное и осязаемое представление системы, которую вы разрабатываете. Они дают осязаемый опыт.

Когда мы перешли от технических заданий к прототипированию, то сразу заметили сокращение количества необходимых уточнений и переделок. Когда-то доля совпадений в интерпретации задания составляла 60–80%; теперь она достигала 90% и даже больше.

Оказалось, на разработку прототипа требуется меньше времени и усилий, чем на создание подробного технического задания и аннотированных каркасных представлений.

Я знаю несколько причин, по которым документация оставляет больше возможностей для неправильного истолкования:

- Техническое задание объемом 60–200 страниц никто не захочет читать. Это не приносит удовольствия.
- Если вы не сможете убедить людей прочесть задание, те не смогут до конца его понять.
- Документация не дает возможности увидеть картину в целом. Приходится читать строчку за строчкой.
- Слова слишком часто можно истолковать по-разному.

Кроме того, прототипы имеют ряд преимуществ, которые помогают снизить вероятность неправильного толкования:

- Вы испытываете работу системы, а не просто читаете о ней.

- Прототипы поощряют игровое поведение. Когда вы даете человеку возможность поиграть с прототипом, повышается вероятность, что тот поймет, как должно работать ваше «детище».

Прототипирование позволяет сэкономить время, усилия и деньги

«У нас нет времени создавать прототип».

«Мы не можем позволить себе прототип. У нас на это нет средств».

Сколько раз вам приходилось слышать такие фразы от клиента, одного из ваших начальников или даже коллег-разработчиков?

Я слышал их десятки раз. Да, они имеют под собой некоторые основания. Разработка прототипа не бесплатна, но его преимущества перевешивают затраты, а главное, потери от неиспользования прототипирования.

Поговорите с любым человеком, перешедшим с процесса проектирования и разработки без прототипирования на процесс с прототипированием, и он скажет вам, что за счет этого он сэкономил кучу времени и избавился от множества проблем. Прототипирование не только дает возможность быстрее реализовать и испытать проект, но и сокращает напрасные потери времени.

Прототипирование сокращает объем напрасного труда

Традиционно в процессе проектирования составляется техническое задание и передается дизайнеру или разработчику. Те изучают требования и создают нечто, основываясь на своем восприятии спецификаций.

Теоретически техническое задание призвано снижать напрасные затраты времени. Главная цель — чтобы все занимались одной и той же задачей. Если все находятся на одном и том же этапе, напрасные затраты времени должны сократиться. Звучит потрясающе.

Правда, только в теории. На практике все иначе. Процесс проектирования и разработки на основе технического задания имеет ряд недостатков, которые и приводят к потерям времени:

1. **Задание написано не для «тех» людей.** Дизайнеры и разработчики редко участвуют в создании технического задания. Его пишет бизнес-аналитик или другой специалист, не имеющий достаточных технических знаний. Поэтому нередко приходится несколько раз переделывать задание.
2. **Слишком много времени и усилий.** Количество времени, потраченного на создание, рецензирование и пересмотр технических заданий, весьма значительно. При разработке сложных систем этот процесс иногда занимает 3–9 месяцев, а иногда и больше. За это время многое может измениться.
3. **Неокончательная финальная версия.** Теоретически техническое задание — окончательный документ. На практике же требования постоянно меняются, даже после того как их составление «завершено».
4. **Неправильное понимание.** Количество неправильно понятых мест в техническом задании объемом 60–200 страниц нередко значительно. непонимание ведет к затратам времени на переделку и отсрочке выпуска продукта.
5. **Несущественные детали.** Техническое задание нередко заполнено описанием малозначимых или незначимых функциональных возможностей. На их создание и тестирование требуется время. Результат — потеря времени на их описание в техзадании, их внедрение и тестирование (часто такие возможности вообще впоследствии не используются).
6. **Слишком позднее обнаружение ошибок.** В техническом задании сложно обнаружить ошибки, пока система не начнет работать. Чем позже ошибка обнаружена, тем дороже ее исправление.

Любой из перечисленных факторов сам по себе приводит к напрасным тратам времени и сил. Но обычно процесс разработки по техническому заданию грешит сразу несколькими из них, что порождает значительную неэффективность. Прототипирование помогает сократить напрасные траты времени и сил и дать следующие выгоды:

1. **Решения принимаются правильными людьми.** Дизайнеры и разработчики могут использовать свои опыт и знания и принимать грамотные решения.
2. **Выживание наиболее приспособленных.** Генерируется и испытывается множество идей, «выживают» сильнейшие из них.
3. **Адаптивность.** Прототипы могут быть доработаны с учетом новых возможностей и требований.
4. **Снижение вероятности неправильного понимания.** Прототип — визуальное, а иногда и физическое представление системы. Его сложнее понять неправильно, чем бумажный документ объемом 60–200 страниц. Снижение вероятности неправильного понимания сокращает количество переделок. Результат — меньшие затраты и более ранний выход на рынок.
5. **Точность.** При использовании прототипирования создаются продукты, более точно соответствующие желаемым характеристикам. Результат — меньшие потери при проектировании, разработке и переделках.
6. **Раннее обнаружение несоответствий.** Прототипирование помогает обнаружить несоответствия на ранних стадиях проектирования и разработки. Чем раньше будут выявлены проблемы, тем дешевле обойдется их исправление.
7. **Снижение рисков.** Прототипирование помогает снизить риски, уменьшая вероятность неправильного понимания и выявляя проблемы на ранних этапах.

Прототипирование не может решить всех проблем разработки по техническому заданию, но способно снизить уровень неэффективности и непроизводительных затрат.

Прототипирование показывает реальные ценности

Джонатан Бейкер-Бейтс — один из тех, кто на опыте убедился в выгодах прототипирования. Он трудится в британской консультационной компании, где используются традиционные методы. Группа разработчиков регулярно получает 200-страничные технические задания, по которым надо определить стоимость и создать продукт. Для этого их и принимают на работу. Компания, где трудится Джонатан, недавно начала использовать прототипирование. Теперь вместо 200-страничного документа они предоставляют высокоточный прототип и 16-страничное описание к нему.

После этих изменений в компании заметили ряд существенных улучшений:

- На разработку прототипа и 16-страничного дополнительного документа требуется меньше времени и сил, чем на написание 200-страничного техзадания.
- Точность оценки стоимости проекта и его продолжительности повысилась на 50%.
- Количество уточняющих запросов от команды разработчиков сократилось на 80%.
- Количество переделок и исправлений ошибок после выпуска продукта уменьшилось до 25% от уровня предыдущих проектов.
- Вся команда согласилась, что прототипирование проще, чем традиционная модель.

Практический пример: прототипирование при жестких бюджете и сроках

Джонатан Бейкер-Бейтс

У нас имелось меньше четырех месяцев на создание «неформального» сайта для одного из ведущих разработчиков компьютерных игр, сильная концепция визуального дизайна, ряд требований к контенту и функциональности и команда, разбирающаяся в CMS. Но бюджет был стесненным, а рамки проекта — неопределенными. К тому же мы в первый раз сотрудничали с этим клиентом, и он был не из тех, кто готов изучать длинные документы (которые нередко доходили до 200 страниц в похожих проектах). Требовалось заинтересовать участников и с самого начала четко определить, что мы собираемся сделать.

Мы решили, что с первого же дня будем разрабатывать функциональный HTML-прототип всего сайта на платформе Axure. Мы надеялись показать почти все необходимые требования с таким уровнем детализации, чтобы они были ясны всем и каждому, будь то CEO* или интегратор CMS.

Конечно, прототип не мог показать все. Нефункциональные элементы, некоторые экраны, появляющиеся при определенных условиях, и исключения приходилось записывать отдельно, так же как и заметки о реализации. Мы вынесли их в дополнительный документ объемом около 20 страниц. Однако в нем были отражены только те моменты, которые явно не показаны в прототипе. Это позволяло предупредить возможность перегруженности документа и его выхода из-под контроля.

Мы сразу обратили внимание на то, что при создании прототипа отпала необходимость в длинных «вводных» описаниях и обсуждениях. Те, кто его видел, понимали, что мы старались сделать, за несколько минут, а не часов или дней. Поэтому мы могли перейти к обсуждению деталей. Это дало нам ряд выгод. Во-первых, точность оценок

* Генеральный директор. Прим. ред.

Практический пример: прототипирование при жестких бюджете и сроках (окончание)

продолжительности работы оказалась очень высокой. Во-вторых, мы потратили на обсуждение проекта (в том числе возможных проблем) в целом примерно на 80% меньше времени, чем обычно. Наконец, интеграция и тестирование прошли гладко, а количество отклонений от спецификаций было примерно на 25% меньше, чем обычно. Бюджетные и временные ограничения никуда не делись, но прототип позволял быстро показать, как сгладить требования, чтобы получить согласие клиента.

Этот подход оказался успешным. Но хочу напомнить читателям, что все проекты разные и в других обстоятельствах этот метод может быть не лучшим решением. Стоит также отметить, что позже, когда сайт начал работу и требовалось постепенно вносить улучшения, мы вернулись к более традиционному процессу, основанному на документах. Но без прототипа мы, скорее всего, не смогли бы уложиться в заданные сроки.

История, рассказанная Джонатаном, не уникальна. Такие же истории я слышал от многих из тех, с кем беседовал во время подготовки этой книги, от десятков людей, посещавших мои лекции о прототипировании, и от участников опросов.

Резюме

Прототипирование действительно помогает сократить сроки разработки на несколько месяцев и даже лет. Так чего же вы ждете?

Теперь вы знаете ценность прототипирования и должны суметь получить согласие на него от клиентов или руководства.

Но не забывайте о следующем:

- прототипирование продуктивно;
- прототипирование дает возможность показать и рассказать;
- прототипирование снижает вероятность неправильного восприятия;
- прототипирование позволяет сэкономить время, усилия и деньги;
- прототипирование создает быструю цепь обратной связи, в результате снижаются риски.

В следующей главе будет рассмотрен быстрый интерактивный процесс прототипирования, которым пользуюсь я.