

Лекция 8.

Прототипирование пользовательского интерфейса: модели и методологии

Цель лекции: в данной лекции рассматриваются вопросы разработки прототипа, их методология. Дается оценка каждому виду прототипа, а также оценка участия пользователя в проектировании путем быстрой этнографии. Описываются все виды рисков при использовании прототипирования.

Вопросы для рассмотрения: Разработка прототипа: методологии и типы прототипирования. Прототипы на бумаге. Контекстное прототипирование. Оценка прототипа. Анализ протокола и модель Нормана. Оценка коммуникабельности. Участие пользователя: быстрая этнография и опытное прототипирование. Средства прототипирования. Прототипирование веб-приложений.

Основные термины: прототипирование, оценка коммуникабельности, быстрая этнография, раскадровка, модель Нормана. риски прототипирования

Введение

Создание прототипа — следующий шаг в разработке пользовательского интерфейса. Прототипирование позволяет создавать макеты интерфейсов разной степени достоверности: от набросков «на скорую руку» и бумажных прототипов до интерактивных макетов с использованием специальных программ.

В английском языке слово prototype (прототип) является существительным или глаголом. Как существительное оно ссылается на первую в своем роде модель, опытный образец или прообраз будущей системы. Как глагол — указывает на процесс разработки дизайна с использованием прототипов (прототипирование). Создание и использование прототипов позволяет дизайнерам проверять их творческие идеи. Почему используют прототипы? Потому что они дешевле, их быстрее создавать, и они могут давать обратную связь.

Помните, что **прототипирование** — именно процесс, а не результат. Вы можете создать экраны и функциональность (или ее имитацию) и назвать это прототипом, но в действительности это лишь часть процесса прототипирования, а не его конечный результат. Результатом прототипирования является обратная связь по концептам, которая может использоваться для улучшения результата.

Разработка прототипа: методологии и типы прототипирования

Процесс создания прототипа обычно состоит из шагов:

- Определение начальных требований
- Разработка первого варианта прототипа, который содержит только пользовательский интерфейс системы
- Изучение прототипа заказчиком и конечными пользователями, получение обратной связи о необходимых изменениях и дополнениях
- Переработка и улучшение прототипа: с учетом полученных замечаний и предложений изменяются как спецификации, так и прототип, после этого шаги 3 и

4 могут повторяться.

Типы прототипирования

- 1) При **быстром прототипировании** предполагается, что создаётся макет, который на каком-то этапе будет оставлен («выброшен») и не станет частью готовой системы. Основное преимущество такого подхода — в скорости: в ответ на свои требования заказчик почти сразу получает прототип интерфейса, и сразу может уточнить требования, до того, как начато написание рабочего кода системы. Стоимость изменения требований на этом этапе очень низкая, поскольку нет кода, который нужно было бы переписывать. Очень важно, чтобы такое прототипирование было выполнено в кратчайшие сроки, поскольку в данном случае тратятся время и ресурсы на код, который не будет в дальнейшем использован.
- 2) **Эволюционное прототипирование** (англ. evolutionary prototyping) ставит своей целью последовательно создавать макеты системы, которые будут все ближе и ближе к реальному продукту. Такой подход имеет то преимущество, что на каждом шаге мы располагаем работающей системой, пусть и не обладающей всей нужной функциональностью, но улучшающейся с каждой итерацией. При этом, не тратятся ресурсы на код, который будет «выброшен». Эволюционный подход к прототипированию может быть выбран, исходя из предположения, что все необходимые требования к моменту начала разработки неизвестны, и будут определяться по мере создания программы; тогда на каждом этапе мы реализуем лишь те требования, которые известны и ясны. Иногда при этом разработчики сосредотачиваются на работе только над теми модулями системы, требования на которые уже определены. В некоторых случаях, когда речь идет о продукте под определенную незанятую нишу, пользователи начинают использовать систему еще до того, как она полностью дописана, в ожидании готовой системы, поскольку «недописанная система — это лучше, чем её полное отсутствие».
- 3) **Инкрементное прототипирование** основано на сборке окончательного продукта из нескольких прототипов. Все части (отдельные прототипы) могут разрабатываться параллельно, тем самым сокращая общее время на разработку.
- 4) **Экстремальное прототипирование** используется при создании веб-приложений. Весь процесс разбивается на три фазы. В первой фазе создается прототип низкой достоверности, состоящий из статических веб-страниц. Во второй фазе создается работоспособный код веб-приложения, а статические веб-страницы переписываются с учетом используемого фреймворка и функциональности, создается полностью работоспособный на уровне модели пользовательский интерфейс. В третьей фазе выполняется интеграция веб-интерфейса со всеми сервисами и ресурсами.

Бумажное прототипирование

Вы имеете всего лишь несколько дней, оставшихся до представления ваших идей относительно приложения. Нет времени на создание полной модели (которая, на самом деле, есть другой прототип). Что вы можете сделать? Вы можете создать бумажный прототип.

Все, что вам потребуется для создания бумажного прототипа обычно есть под рукой:

- картон, линованная и нелинованная бумага, разноцветная бумага для заметок (sticky-notes), листы прозрачной ацетатной пленки;
- цветные ручки, фломастеры и карандаши;
- канцелярский клей, клейкая лента, клей многократного применения (подобный тому, который наносится на клейкие заметки);
- ножницы, канцелярский нож, хорошая линейка, циркуль.

Вы можете не иметь всех перечисленных принадлежностей, но стоит собрать большинство из них, прежде чем начнете.

При бумажном прототипировании можно мгновенно модифицировать результат, то есть если потенциальный пользователь находит явную ошибку в вашем творении, вы можете легко внести исправления, прежде чем показать результат следующему пользователю.

Контекстное прототипирование или раскадровки

Раскадровка — это последовательность зарисовок, показывающих, как пользователь продвигается «сквозь» задачу, используя конкретное устройство. Это могут быть эскизы графического пользовательского интерфейса (GUI), скриншоты или наброски сцен пользовательского взаимодействия с программой или устройством. Раскадровки очень хороши для «оживления» сценариев взаимодействия.

Для того, чтобы создавать раскадровки интерфейса не обязательно быть художником. Здесь не нужна фотореалистичность, поскольку каждая сцена раскадровки является, в первую очередь, визуальным средством общения. Концентрируйтесь на общении, а не на красоте рисунка. Это должно помочь

вам думать о раскадровке, как о совокупности пиктограмм, каждая из которых обозначает объект или действие. Ни что не мешает вам нарисовать пользователя в виде фигурки из черточек, а компьютер — в виде прямоугольника.

Оценка прототипа

Основными преимуществами прототипирования являются сокращение времени и стоимости разработки за счёт того, что оценка прототипа позволяет на более ранних стадиях обнаружить недостаточность или несоответствие требований. Чем позднее проводятся изменения в спецификации, тем они дороже, поэтому уточнение «чего же пользователи/заказчики хотят на самом деле» на ранних стадиях разработки снижает общую стоимость.

Психологически важную роль играет также вовлечение заказчика в процесс разработки. Работа с прототипом позволяет будущим пользователям увидеть, как будет выглядеть будущая программа, и повлиять на её поведение, что уменьшает расхождения в представлении о программе между разработчиками и пользователями. Снижается и эффект почти неизбежного отторжения новой системы при внедрении, особенно когда она внедряется на место ранее использовавшейся.

Тем не менее, использование прототипирования создаёт ряд дополнительных рисков.

- ✓ **Риск недостаточного анализа.** Концентрация усилий на ограниченном прототипе может отвлекать разработчиков от надлежащего анализа требований на полную

систему.

- ✓ **Риск смещения прототипа** и готовой системы в представлении пользователей. Пользователи могут подумать, что прототип, который в действительности предполагается «выбросить», и есть основа будущей системы. Это может привести к двум противоположным по сути негативным эффектам. Во-первых, пользователи могут ожидать от прототипа более точного поведения и, не обнаружив его, разочароваться в возможностях разработчиков. Во-вторых, наличие быстро созданного прототипа может создать впечатление того, что «почти вся работа уже сделана», что может стимулировать попытки неоправданно сократить время и/или бюджет разработки.
- ✓ **Риск потери времени** на создание прототипа. Ключевое свойство прототипа — то, что он создается быстро. Если разработчики тратят время на создание слишком сложного и функционального прототипа, они теряют преимущества от применения прототипирования вообще.

Анализ протокола и модель Нормана

Анализ протокола заключается в помещении респондента в ситуацию принятия решения о покупке, в процессе которого он должен подробно описать все факторы, которыми он руководствовался при принятии этого решения.

Респонденту предлагают представить себе конкретную ситуацию, в которой он должен принять решение. После этого он должен словесно описать все факторы и аргументы, которыми он руководствовался в процессе принятия решения. Иногда при применении данного метода используется диктофон. Затем исследователь анализирует протоколы (отсюда - название метода), представленные респондентами.

Психологи Норман и Во в своих трудах первыми осветили модель памяти, основанную на двойственности. Эта модель стала базой для многих современных теорий и исследований в области механизмов и структуры сквозного психического процесса под названием память. Ученые разделили память на первичную и вторичную. В 1965 году была представлена модель, дополненная такими понятиями, как информационные ячейки (в современной когнитивной психологии это понятие получило название «ящик в голове»).

Ученые в своих экспериментах преследовали цель дать количественную оценку свойствам первичной (кратковременной) памяти. Придерживаясь основного принципа ограниченности первичной памяти, они изучали механизмы стирания информации, ее исчезновения, забывания.

Выводы, к которым пришли Норман и Во были следующими: информация, хранящаяся в ячейках блока кратковременной памяти хранится короткое время не потому, что срок ее хранения ограничен временными интервалами, а потому, что она вытесняется новой информацией, поступающей в сенсорные регистры. Здесь имеет место принцип интерференции, т.е. вытеснение старого материала новым.

Ученые также выделяли кратковременный и долговременный блоки памяти, выявляя процессы закономерности перехода из первичного хранилища во вторичное. Метод повторения, имеющий место быть в их экспериментах нес три базисные функции перехода из блока первичной памяти в долговременную:

1. *Соппротивление затуханию следов в ячейках кратковременной памяти,*
2. *Предохранение материала от эффекта интерференции,*

3. Перевод информации из первичного во вторичное хранилище – фиксация информации.

Оценка коммуникабельности

Утверждалось, что прототипирование, в той или иной форме, должно использоваться постоянно. Тем не менее, прототипирование наиболее полезно в системах, которые будут иметь много взаимодействия с пользователями.

Прототипирование оказалось очень эффективным при анализе и проектировании онлайн-систем, особенно для обработки транзакций, где использование диалоговых окон на экране является гораздо более обоснованным. Чем больше взаимодействия между компьютером и пользователем, тем больше пользы можно получить от построения быстрой системы и предоставления пользователю возможности играть с ней.

Системы с небольшим взаимодействием с пользователем, такие как пакетная обработка или системы, которые в основном выполняют вычисления, приносят мало пользы от прототипирования. Иногда кодирование, необходимое для выполнения системных функций, может быть слишком интенсивным, а потенциальные преимущества, которые могут обеспечить прототипирование, слишком малы.

Прототипы особенно хороши для разработки хороших человек-компьютерных интерфейсов. "Одно из наиболее продуктивных применений быстрого прототипирования на сегодняшний день уже используется в качестве инструмента для итеративного проектирования пользовательских требований и проектирования человеко-компьютерного интерфейса."

Участие пользователя: быстрая этнография и опытное прототипирование

Понимание того, для какой целевой аудитории создается прототип и с какими намерениями, определяет все остальные аспекты этого процесса. Выявив аудиторию и ее намерения, вы сможете сделать следующее:

- определить, что необходимо для создания прототипа;
- установить приемлемые границы ожидаемого;
- определить требуемый уровень точности;
- выбрать правильный инструмент для работы.

Для начала нужно решить вопрос о целевой аудитории – это отправная точка разработки. Когда вы поймете, кто будет изучать прототип, то сможете определить, что от него требуется, насколько подробным должно быть прототипирование и насколько высокая точность желательна.

Если вы не умеете или не можете писать код, есть несколько способов симитировать этот процесс:

- Можно сформировать серию экранов в формате JPEG. Используйте Dreamweaver для создания изображений и связывания их друг с другом. Вы не напишете ни строчки кода, но сможете получить отзывы об интерфейсе и понять, если ли в идее смысл.
- Применяйте встроенные функции Fireworks для связывания страниц и фреймов и создайте HTML-прототип, с которым можно работать с помощью мыши.

- Используйте Adobe Acrobat или другой редактор PDF и свяжите экраны между собой, чтобы добиться такого же результата, как и в предыдущем случае.
- В PowerPoint вы сможете связать последовательность статичных экранов.

Средства прототипирования

HTML является цифровым аналогом бумажного прототипа. Он (иногда) бесплатен и прост. Даже если вы не обладаете мастерством HTML-гуру и программиста веб-интерфейсов, знания основ HTML вполне достаточно для успешного создания HTML прототипа.

Пожалуй, самый дешевый способ создания простого HTML прототипа «на скорую руку» - кодирование вручную в текстовом редакторе.

Power Point и Keynote

Прототипы Power Point и Keynote относятся к категории «дешево и сердито», но иногда ничего большего и не требуется. Прототип Power Point или Keynote строится так же, как обычная слайдовая презентация, с простейшими интерактивными возможностями. Обе программы позволяют организовать интерактивное взаимодействие для имитации той последовательности операций, которую вы хотите проверить на своих пользователях. Опытные пользователи Power Point или Keynote могут даже встроить анимацию и другие элементы, повышающие интерактивность прототипа.

Прототипирование веб-приложений

- **Axure RP Pro**

Инструмент, ориентированный на создание прототипов веб-сайтов. Генерирует кликабельный HTML и документацию в формате Word. Поддерживает комплексное взаимодействие. Windows / \$ 589

- **EasyPrototype**

Очень похож на популярный Axure, легкий инструмент, позволяет проектировать экранные формы, экспортировать интерактивные HTML- прототипы и документацию. Кроссплатформенный / \$ 69

- **Fireworks**

Возможно создание сложных интерактивных прототипов. Множество инструментов аналогичны некоторым инструментам из Adobe suite. Имеется возможность экспорта в PDF или HTML. Кроссплатформенный / \$ 299

- **JustProto**

Веб-приложение, ориентированное на работу с удалённой командой. Веб / \$ 99

Вопрос о том, в какой мере оправдано применение прототипирования, в той или иной форме, к конкретному проекту, не имеет однозначного ответа. Хорошие результаты дает прототипирование при проектировании систем, в которых заметная часть трудозатрат связана с реализацией интерфейсов человек-компьютер, а оценка соответствия системы требованиям заказчика в значительной мере субъективна. В таких системах создание детальной спецификации очень сложно, так как заказчик не может учесть все нюансы, а

программисты организуют неспецифицированные детали по собственному усмотрению; в результате типичной реакцией на демонстрацию готовой системы является: «Да, всё формально соответствует техническому заданию, но нам было нужно совсем не это!». Создание и оценка прототипа позволяют в таких случаях на ранних стадиях разработки отделить важные для заказчика детали, которые не попали в спецификацию, от неважных, реализация которых может быть произвольной.

Контрольные вопросы

1. Что такое прототипирование проектирования пользовательского интерфейса?
2. Какие типы прототипирования существуют? Объясните сущность каждого из них.
3. В чем заключается модель Нормана?
4. Какие риски прототипирования существуют и можно ли их исключить?
5. Какие средства относятся к видам прототипирования?
6. Дайте классификацию прототипов, используемых в юзабилити-тестировании.
7. В каких ролях выступают участники тестирования в методе бумажных прототипов?
8. Какие проблемы выявляют с помощью метода бумажных прототипов?
9. Разработайте сценарий компьютерного варианта метода карточной сортировки.

Литература:

1. Купер А. т.б. Об интерфейсе: основы проектирования взаимодействия. 4-е изд. – СПб: Питер, 2018. – 720 б.
2. Норман Д. Дизайн привычных вещей. – М: Манн, Иванов и Фербер, 2019. – 384 б.
3. The Encyclopedia of Human-Computer Interaction, 2nd Ed. [электрондық басылым] – URL: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed> (соңғы алышқан 8/1/2020)
4. Sharp Н. Interaction Design: Beyond Human-Computer Interaction. 5th Ed. – Wiley, 2019. – 656 p.
5. Shneiderman В. et.al. Designing the user interface: strategies for effective human-computer interaction. 6th Ed. – Pearson, 2016. – 616 p.
6. Тидвелл Д. т.б. Разработка пользовательских интерфейсов. 2-е изд. – М: Питер, 2011 – 480 б.
7. Круг С. т.б. Веб-дизайн или «не заставляйте меня думать». – СПб.: Символ Плюс, 2008. – 224 .
8. Нильсен Я. т.б. Веб дизайн. СПб: Символ Плюс, 2006. – 512 б.
9. Уильямс Р. т.б. Не дизайнерская книга о дизайне. – СПб: Весь, 2004. – 128 б.
10. <http://appcamp.io/> – Онлайн-курс который дает начальное понимание разработки на HTML и мобильных платформах.
11. <http://phonegap.com/book/> – Список книг по разработке HTML и мобильных приложений с помощью фреймворкаPhoneGap.
12. <http://creator.ionic.io/> – HTML фреймворк пользовательского интерфейса для мобильных приложений.