

ЧТО ДЕЛАТЬ, когда даже Agile «не рулит»?

Часть II

Итак, уважаемый читатель, как это было описано в первой части данной статьи, «на входе» имеем набор идей, весьма далёких от уровня *software functional requirements* и заказчика (его представителя, ответственное лицо и т. д.), не склонного к кропотливой работе над выявлением требований и искренне полагающего, что его участие в разработке должно ограничиваться лишь оценкой соответствия конечного продукта собственным переменчивым желаниям.

Требуется из абстрактных идей и разрозненных бизнес-артефактов получить конкретные функциональные требования к ПО, на основании которых можно будет проектировать архитектуру и писать код, при этом, не упуская скрытую функциональность и не перегружая заказчика утомительным чтением спецификаций и сценариев, а также бесконечными опросами встречами и интервью.

Используем для этого методологию реверсивного анализа требований на основе динамических html-прототипов, создаваемых с помощью Axure RP 7.0.

В самых общих чертах процесс можно представить следующим образом:

1. Выявляем пользователей и высокоуровневые пользовательские требования, записываем их в форме кратких сценариев или user story и согласуем их с заказчиком и располагаем в порядке их приоритета.
2. В Axure RP создаём проект, в котором предусматриваем страницы для ролевого разграничения пользователей (Рис. 3) и навигации по функциональным сервисам (Рис. 4).
3. Реализуем очередное по списку требование на отдельной странице прототипа, добиваясь по возможности полной имитации его функционирования. Для этого используем виджеты, их библиотеки (в том числе собственные), применяем динамические панели, вводим глобальные и временные переменные, описываем условия и события. При этом отрабатываем как типовые, так и альтернативные потоки взаимодействия.
4. Тестируем страницу в html (меню "Publish-Preview" в Axure RP Pro 7.0), оцениваем полноту реализации функционала, возможности ошибочных действий пользователя, удобство и интуитивность интерфейса.
5. При необходимости вносим изменения в базовую функциональность, добавляем в прототип функциональные элементы, исключая ошибочные действия пользователей и улучшающие UX, а также вводим необходимые текстовые инструкции и пояснения.
6. Генерируем проект в «облаке» Axure RP под уникальным именем, посылаем ссылку заказчику и предлагаем протестировать разработанную часть функционала. В случае, если сценарий взаимодействия сложный и интуитивно не понятный пользователю, то прилагаем видео-инструкцию (в виде файла .mp4, созданного, например, с помощью продукта Snagit), или краткую инструкцию, выполненную с помощью стандартных средств MS Office.
7. Общаемся с заинтересованными сторонами (Skype, телефон) и по результатам вносим в прототип необходимые изменения, дополняем UML-диаграммами, после чего повторно генерируем html-страницы и переходим к разработке следующего по списку сервиса.
8. По завершению прототипирования создаём общую для всего проекта спецификацию функциональных и нефункциональных требований.

Примечание: понятие функциональный сервис в данном случае – это то, что должна делать программная система во взаимодействии с пользователем с целью достижения значимого для него результата. Другими словами, это вариант использования (Use Case) по определению Айвара Якобсона (один из создателей языка UML, который ввел это понятие в 1986 году).

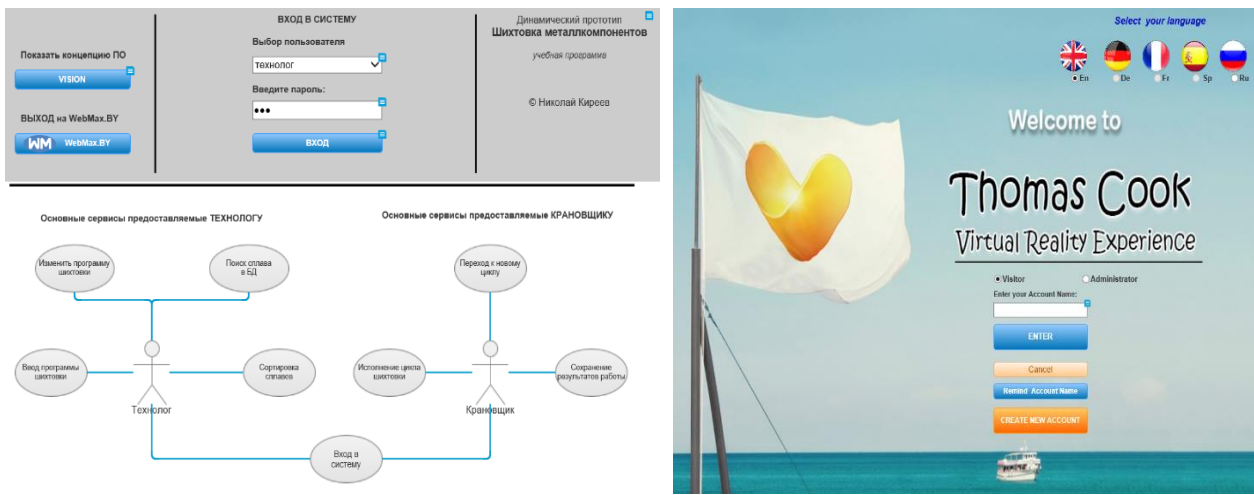


Рисунок 1. Примеры страниц с разграничением доступа



Рисунок 2. Пример страницы функциональных сервисов

Теперь подробнее о функциях, отражаемых в прототипах, касательно представленной в первой части статьи иерархии (Рис.2).

Предположение: в прототипе можно за редким исключением имитировать всё, что будет отражаться на экранах программных приложений, включая мобильные.

Имитируются функции front-end-уровней: списки вносятся, изменяются, удаляются, сортируются, выделяются строки, производятся расчёты, обеспечиваются переходы по условиям, обеспечивается доступность и видимость элементов интерфейса, показываются и убираются комментарии и т.д. Основной механизм для создания имитации – динамические панели, отличающие Axure RP от аналогичных средств, позволяющих создавать вайрфреймы, мокапы и прототипы с ограниченной динамикой.

Функции, отнесённые к back-end тоже могут имитироваться в прототипе, если их результат отражается на экране пользователя. Это может делаться с помощью искусственных приёмов, не соответствующих реальным условиям. Например, в проекте «Шихтовка металлкомпонентов» [7] для получения веса компонента, находящегося на кране, необходимо кликнуть в текстовом поле «Вес на кране» (Рис. 5), после чего произойдёт имитация получения протокола через порт RS232 и в указанном поле появится заранее внесённое в прототип значение. Имитация запроса и получения добавок горения, а также действий системы с ними связанных организована с помощью внутреннего таймера: кликаем кнопками запросов, через определённое, установленное в прототипе время видим реакцию системы, при этом никакие протоколы не посылаются и не принимаются.

КОНТРОЛЬ ПРОПОРЦИИ МЕТАЛЛКОМПОНЕНТОВ

Наименование сплава: ЛЮТИК-5 Дата: 21.5.2014 Вес сплава: 200

СПИСОК МЕТАЛЛКОМПОНЕНТОВ

№	Наименование МК	Заданный вес, кг	Остаточный вес, кг	Пропорция, в %
1	стружка пресс.	200	200	20
2	железо кровельное	550	500	30
3	заготовка стальная	950	950	50

КОНТРОЛЬ ДОБАВОК

ЗАПРОС КОКСА ИЗВЕСТИ: Вес кокса, кг: 125 Вес извести, кг: 64 Вес кокса и извести, кг: 189

ЗАПРОС СПЕЦКОКСА: Вес спецкокса, кг: 87

РАЗГРУЗКА ТРАНСП.:

ЛЕГИРУЮЩИЕ ДОБАВКИ

FeSi: 0 FeCr: 0 FeMn: 0

ВЕС НА КРАНЕ: 50

СУММАРНЫЙ ВЕС МК: 50

Заполнение контейнера: ЗАПОЛНЕН

ОКНО СООБЩЕНИЙ

Спецкокс получен. Клавишей РАЗГРУЗКА ТРАНСП разгрузите транспортер

Рисунок 3. Пример имитации работы приложения «Шихтовка МК»

Достоинства методики

1. Спецификация требований создаётся на основе действующего протестированного и согласованного с заказчиком прототипа, функционал которого гораздо ближе к реальной программной системе, чем любая написанная до начала разработки спецификация или визуальная модель.
2. Обеспечивается необходимая и достаточная полнота функциональных требований. Выявляются скрытые функции, а также оценивается необходимость ввода сервисных функций, связанных с удобством использования и исключением ошибок пользователей.
3. Заказчик может заранее и без больших затрат оценить достоинства и недостатки программной системы и эффективность достижения поставленных им бизнес-целей.
4. Разработка производится итерационно при постоянном взаимодействии и под контролем заказчика, который в любое удобное для него время может тестировать прототип, постоянно находящийся в «облаке».

Заключение

Важный вопрос, который, волнует всех, мужественно выдержавших до этого момента: сколько времени потребуется на создание прототипа и будут ли оправданы затраты на его разработку?

Всё зависит от умения, опыта работы в Axure RP и от наработанных библиотек. В начале её изучения, более-менее быстро делаются вайрфреймы и навигационная динамика, но по мере накопления опыта в скриптах, создании и использовании переменных, в управлении динамическими панелями, имитация занимает всё меньше времени и значительно выигрывает по сравнению с кодированием, что особенно существенно при многочисленных изменениях в функциональности.

При очевидных преимуществах методики, время, затраченное на разработку требований, в целом увеличивается, но не на столько, чтобы отказываться от возможностей динамических прототипов. Подготовка статических вайрфреймов в комплекте со сценариями занимает не на много меньше времени, но главное, не позволяет в полной мере выявить скрытые функции, обеспечивающие бизнес-логику, а также обеспечить предотвращение ошибок и удобство работы пользователей. Кроме того, заказчики предпочитают работать с функционирующим продуктом, чем с подробными описаниями его функционирования. Многие из них с удовольствием готовы тестировать, нажимая на кнопки, внося данные, а затем визуально оценивать результат, чем внимательно вчитываться в тексты и включать своё воображение.

Перед началом работ с заказчиком можно оформить предварительный договор на разработку действующего прототипа и спецификаций требований. Используя прототип, он может решить на сколько ему полезен тот или иной функционал (в соответствии с моделью Нориаки Кано [8]), а разработчики смогут более точно оценить сроки и стоимость проекта.

Многие аналитики возразят, что вторжение в область UX не является их «епархией», но как тогда относиться к тому, что одним из требований к Scrum-командам, является кросс-функциональные возможности их участников? Следует заметить, что только очень крупные ИТ-фирмы могут позволить себе узкопрофессиональный персонал, чем мельче фирма-разработчик, тем острее потребность в кросс-функциональных специалистах.

Описанный выше процесс разработки прототипа отдалённо напоминает Scrum и тем, что заказчик может ежедневно контролировать разработку и оперативно вносить необходимые изменения. При этом не требуются ежедневные standup'ы в строго определённое время, прототип, расположенный в «облаке» может быть доступен в любое удобное время и даже с мобильного устройства.

При необходимости, Axure RP позволяет сгенерировать подобие спецификации в формате MS Word. Этот документ может использоваться, но потребует значительной переработки.

Большим плюсом является ещё то, что найденные решения удобно сохранять в библиотеке для дальнейшего использования в других проектах. В библиотеке должны храниться исходные файлы формата Axure RP и их html-прототипы, наподобие данного примера [9].

Использование готовых библиотек существенно ускорит процесс создания динамических прототипов.

Буду рад ответить на вопросы и принять участие в обсуждении материалов статьи.

Все желающие попробовать методику реверсивного анализа с использованием динамических прототипов на небольших практических примерах, приглашаются принять участие в online-вебинарах.

Подробности на <http://www.webmax.by/vebinary/>, условия и запись на <http://fy7qtu.axshare.com/#c=2>

Ссылки на источники:

7. Учебная программа <http://www.webmax.by/docs/shicht/home.html>
8. Модель Н. Кано <http://www.fdfgroup.ru/?id=281>
9. Пример html-прототипа <http://ygz4b5.axshare.com/#c=2>.

Автор:



Николай Киреев, старший преподаватель ИИТ БГУИР, индивидуальный предприниматель (разработка ПО, студия **WebMax.BY**), руководитель проектов, аналитик (freelancer)

Skype: nousy123,
e-Mail: nousy@mail.ru