

Разработка: лицом к клиенту

Алексей Чумаков

Выступление
для семинара UPA Russia, 2007

План

1. Как выглядит вынужденная разработка?
2. Что при этом происходит?
3. Как построить по-другому?
4. Резюме

1. Как выглядит вынужденная разработка?

Часто разработка ВЫГЛЯДИТ ТАК:

- приходит заказчик
- говорит: хочу сайт... вот вам мои представления, вот структура... вот контент...
- мы согласуем ТЗ и договор, включаем все, что сказал заказчик...
- ударяем по рукам... делаем...
- показываем заказчику результат —
- заказчику не нравится — и он говорит: «переделать!»

И у нас есть выбор:

- «сдать» по букве договора, получить деньги... и потерять недовольного заказчика
- переделать работу за свой счет, чтобы его сохранить... и потерять деньги

Можно так...

- Можно, конечно, работать, как «фабрика одного заказа». **Заказчиков много, другие придут!**
- цена продажи — все выше и выше
- «а осадок остался» — и работа становится вынужденной.

Не хочется?

Что при этом происходит?

Заказчик...

Заказчик, держа в голове потребность, представляет ценность, которая ее удовлетворяет:

- Хочу пить — ищу воду...
...или шампанское в ресторане...
...или ...
- Хочу (что-то?) — заказываю веб-систему — описываю структуру, цвет, текст...

Что и где заказать — определяет исходя из своего опыта.

Подрядчик...

Подрядчик принимает заказ — и начинает делать... веб-сайт. Или приносит шампанское.

- Но вода может быть нужна не только, чтобы напиться, но и чтобы умыться.
- Подрядчик считает, что заказчик «знает, что говорит» — считая его профессионалом, как себя.
- Заказчик молчит о том, **зачем** ему нужен сайт — для него это само собой разумеется.

Исходная потребность всегда подразумевается и редко произносится.

Усложняем картину

- Приходит с заказом часто не тот, у кого есть исходная потребность, а **исполнитель (посредник)**, который хочет, чтобы...

*«у него все было, и ничего за это не было»
— похвалили при минимуме затрат.*

Что дальше?

- **Исполнитель** делает и показывает... заказчику не нравится.
- **Заказчик** осознает, что исполнитель не понимает его потребности... «исполнитель – полный идиот», думает он...
- ...и из лучших побуждений, считая, что деньги уже заплачены, и деваться некуда, начинает объяснять ему, как маленькому, какие детали ему не нравятся. «Здесь окошко передвинуть, здесь цвет не тот...»
- **Исполнитель** понимает, что заказчик держит его за идиота — и сопротивляется, несогласный, но вынужденный подстраиваться, чтобы получить свои деньги.

Итог такой работы

- исполнитель зол на заказчика (тот давит и держит за идиота!)
- исполнитель недоволен своей работой (кому нравится, когда все время ругают?)
- исполнитель работает «спустя рукава». Бюджет потрачен, да и зачем лучше?
- заказчик зарекается обращаться к такому исполнителю (если есть другие)

Разработчик мучает себя и клиента
(Почему только разработчик? потому что он профессионал, и это его ответственность)

Не нравится?

Как построить по-другому?

Метафора ресторана

Разработчик — ресторан, где есть

- ◆ зал со столиками, официантами, музыкой...
- ◆ кухня с поварами
- ◆ МЕНЮ
- ◆ ВЫВЕСКА

Шаг 1: КТО заказчик?

Мы должны понять, **кому** нужна разработка?
Кто истинный заказчик?

Ресторан:

- ◆ Пришел ли «курьер, купить обед для начальника»?
- ◆ Молодой человек, чтобы произвести впечатление на девушку (девушка)?

Веб-разработка:

- ◆ директор по продажам — веб-сайт для привлечения клиентов?
- ◆ первое лицо — веб-сайт, чтобы круче, чем у конкурента?

Проверяем себя: *Он оценивает? Его потребность реализуется? Как именно? Он платит?*

Шаг 2: КТО пользователь?

Если пользователь недоволен — заказчик будет тоже недоволен.

Выясняем, *кто* пользователь:

Ресторан:

- ◆ Начальник?
- ◆ Девушка?

Веб-система:

- ◆ Клиенты компании?
- ◆ Сотрудники-операторы?

Проверяем себя: *ему есть зачем* пользоваться системой.

Пользователь ≠ заказчик!

Заказчик склонен забывать про пользователя!

Шаг 3: ЗАЧЕМ им это?

И пользователь, и заказчик считают свою цель очевидной.

И автоматически считают, что эта цель *очевидна для вас*.

Задача разработчика: выявить, **зачем** нужна разработка?

- ◆ спросить у заказчика / пользователя
- ◆ представить самим и подтвердить у него

Проверяем себя: **личная выгода** (получит ли этот потребитель лично выгоду? какую?)

Шаг 4: ЧТО и КАКОЕ им надо?

- Переформулируем запрос заказчика и пользователей
- Описываем продукт исходя из выявленных потребностей
- Согласуем с заказчиком и подписываем договор

Что:

- название и назначение системы
 - ◆ «борщ»
 - ◆ «портал для корпоративного обучения»

Какое:

- борщ — ...
- портал — названия и краткое описание сценариев, прототипы...

Проверяем себя: ему **этот продукт** нужен? Он **готов заплатить за это?**

Шаг 5: делаем

- делаем именно то, о чем договорились
- все время помним и следим: **кому? что? какое? зачем?**
- часто показываем заказчику (лучше — не реже чем 1 раз в неделю)
- пересогласовываем, если отклонились от **кому? что? какое? зачем?**

Если все правильно:

- ◆ заказчик активно заинтересован и оценивает, **что мы делаем**

Признак того, что мы делаем не то:

- ◆ **заказчик замолкает и теряет интерес**
- ◆ **заказчик объясняет, как нам делать!**

Тогда возвращаемся назад: **кому? что? какое? зачем?**

Как защитить «кухню»?

Заказчик часто вмешивается, говоря, *как именно сделать*, например, какую технологию выбрать... Почему?

- **Считает, что неправильно делаем:**
 - ◆ спрашивать
 - ◆ раньше и чаще давать обратную связь («готовим блюда на глазах у клиента»)
- **Хочет сам участвовать в процессе**
 - ◆ вовлечь (модель «приготовь блюдо сам»)
- **Не наш клиент... «пришел за суши в мексиканский ресторан»...**
 - ◆ «сбегать за суши в за угол» — заказать у партнера
 - ◆ повесить вывеску и написать меню (какой мы ресторан? какие блюда?)
 - ★ системы и технологии (специализация)
 - ★ цены
 - ★ условия

Разработка: вид для заказчика

У заказчика есть образ системы — строим разработку вокруг прототипов с постепенной детализацией:

- **Отвечаем на потребность заказчика** — формулируем название системы и названия *сценариев-функций* (или требований) системы (**что? какое?**)... согласовываем с заказчиком
- **Отвечаем на потребность пользователя** — описываем сценарии и прототипы... проводим юзабилити-тестирование, согласовываем с заказчиком
- Реализуем систему
- Общаемся с заказчиком и пользователями

Разработка: вид для технолога

Реализация системы — это кухня ресторана. Главный на ней — шеф-повар, и он определяет, что и как готовить.

Именно здесь определяется удовольствие от разработки!

Инструменты:

- ◆ спецификация архитектуры системы (САС) — рецепт
- ◆ общие библиотеки... — полуфабрикаты
- ◆ правила оформления... — ТУ
- ◆ проверка качества...

Для заказчика технология обычно не важна!

Хорошая технология — это способ извлечь максимум прибыли в пользу разработчика!

Довольный заказчик склонен простить непрожаренный бифштекс!

Недовольный — всегда найдет причину для придирки.

Общение с заказчиком

- В хорошем ресторане официант и администратор следит за чистотой столиков, за вниманием к клиентам
- В плохом — официант швыряет тарелку на стол клиенту, разворачивается и уходит.

Многие подрядчики выдают: «вот, мы сделали по ТЗ. принимайте». Какой это тип официанта?

Как преодолеть в себе? Тренинги личностного роста.

А какова ваша разработка?

Общение с заказчиком 2

Не реже, чем раз в неделю — показываем клиенту систему и задаем вопросы...

- ◆ *всем ли он доволен?*
 - ◆ *что он еще хочет?*
 - ◆ такая ли система нужна?

 - ◆ нужна ли заказчику и пользователям такая система? тем ли людям она нужна? есть ли еще кто-то?
 - ◆ удовлетворяет ли система потребностям участников?
- **Не боимся**, что заказчик захочет что-то еще за те же деньги: довольный клиент платит больше — надо лишь вовремя выставить счет!

Результат:

- заказчик получает больше, чем продукт — он рад вниманию и удовлетворению
- заказчик дает больше, чем расчет: он возвращается и рекомендует другим

При чем тут юзабилити?

Usability — это пригодность для использования.

Пригодна для заказчика и пользователя *только та* система, которую тот *хочет* получить.

Разработчик же часто добивается обратного...

Резюме

- Метафора ресторана
 - ◆ вывеска и меню
 - ◆ внимательный официант
 - ◆ повар вкусно готовит
 - ◆ кухня отдельно от зала
 - ◆ приготовление при клиенте / вместе с клиентом
- Удовольствие для заказчика:
 - ◆ шаг первый: выявить, **кто** заказчик?
 - ◆ шаг второй: выявить, **кто** пользователи?
 - ◆ шаг третий: узнать, **зачем?** какие потребности удовлетворяет?
 - ◆ шаг четвертый: согласовать, **что и какое** им надо?
- Удовольствие для разработчика:
 - ◆ шаг пятый: оставить себе то, **как** делается разработка.

Довольный посетитель прощает непрожаренный бифштекс...

...довольный заказчик веб-разработки способствует удовольствию от самой разработки, и не учит «повара готовить».

Алексей Чумаков

сейчас:

- консультант по организационному развитию

ранее:

- директор департамента ИТ-аутсорсинга, «Риал»
- директор департамента ИТ «Инком-Недвижимость»
- руководитель проектов как «от заказчика», так и «от подрядчика»
- эксперт Borland Intl.
- бизнес-консультант

опыт:

- разработка информационных систем
- организация проектного офиса
- предоставление ИТ-услуг
- создание ИТ-инфраструктуры
- организация сетей связи
- управление сервисными подразделениями
- кризисное управление проектами
- более 100 реализованных проектов

Вопросы?

Разработка: лицом к клиенту

Алексей Чумаков
alex@chumakov.ru