

Как мы узнали SCRUM

История большой любви
дружбы Заказчика и
Подрядчика



Вводная - Заказчик



- Мы – второй подрядчик, взявшийся делать проект, продолжительность предыдущего проекта – 2 года.
 - E-commerce solution – низкий уровень удовлетворения
 - CMS Solution – низкий уровень удовлетворения
 - **Предыдущий РМ – высокий уровень неприятия**
 - Допущенные ошибки – исправленные самим заказчиком дали ему повод считать что он умнее
 - Сложившийся стиль работы – «орать-и-плющить»
- **Как результат – сложный клиент, предубежденный против компании-разработчика, с тяжелым стилем общения**

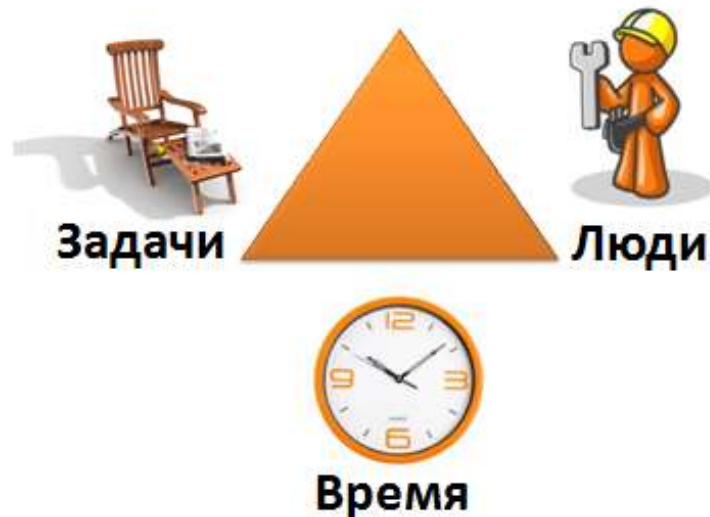
Вводная - проект

- Система продаж, интегрированная с SAP R/3. Охватывает весь цикл продаж, начиная с заказа и заканчивая отгрузкой. Приложение .NET взаимодействующее с SAP через Business Connector.
- Объективные сложности, встреченные сразу:
 - Менее 3 месяцев на реализацию проекта с незнакомой технологией
 - Интеграция с системой, находящейся в стадии разработки
 - Недокументированные и нетестированные API
 - Отсутствие возможности тестирования ПО нашими специалистами
 - Отсутствие аналитики – клиент ориентировочно представлял, ЧТО он хочет, но очень слабо представлял себе КАК это должно быть
- Очевиден Agile, но как делать Agile за fixed price? 😊

Начало проекта

- Проект – fixed price. Прошлая попытка – 22 месяца разработки при оплаченных бти.

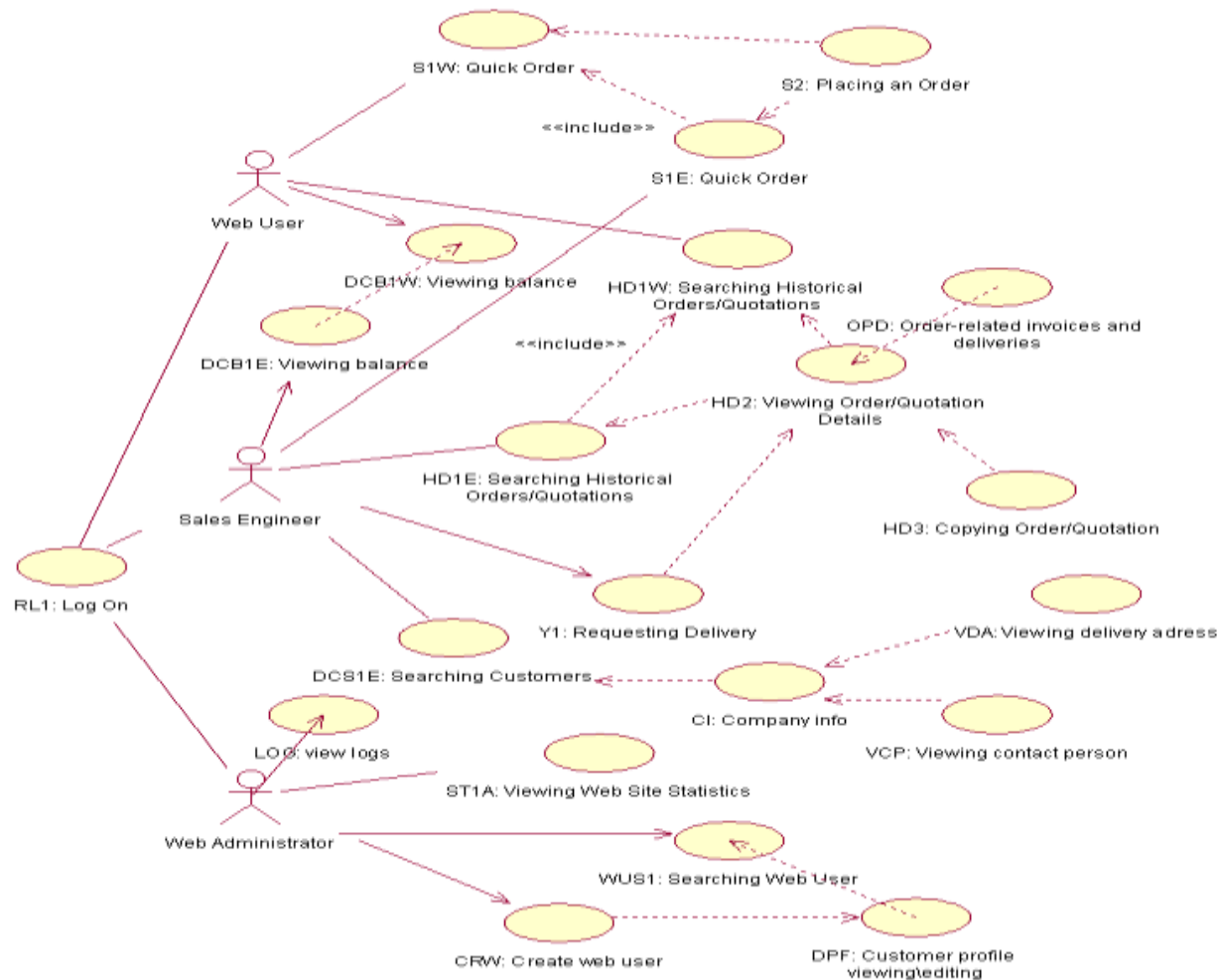
Чтобы избежать подобной ситуации:



1. Четко обозначили границы проекта – отсекли CMS, интеграцию и пр.
2. Просмотрели что система должна делать – описали процессы в системе
3. Попытались сразу сформировать как можно более полное видение того, как приложение будет выглядеть и взаимодействовать с пользователем – используя прототипы
4. Сняли с себя ответственность за ограничения производительности, связанные с SAP BC
5. Зафиксировали это все документально

Фиксирование границ

These use-cases are described in "Use Case Specifications.doc"



Пилотный проект

- Унаследованный прототип – «быстрый заказ», «корзина», «размещение заказа». Прототип системы не работал даже спустя 3 недели.
- Не работал по следующим причинам:
 1. «Кривые» примеры. Примеров входного-выходного XML не хватало, и они не охватывали все возможные ситуации. Кроме того, менялись настройки SAP.
 2. Заказчик описывает основной сценарий, но не описывает ветвления.
 3. Отсутствие тестирования. Изучая новую технологию, нужно иметь возможность все время проверять свои предположения.
 4. Плохой код. Код писался как попало и в конце явно было видно как надо было делать.
- Приняли сложное решение и убедили заказчика его принять – выбросить прототип и реализовать все три кейса заново

Организация процесса

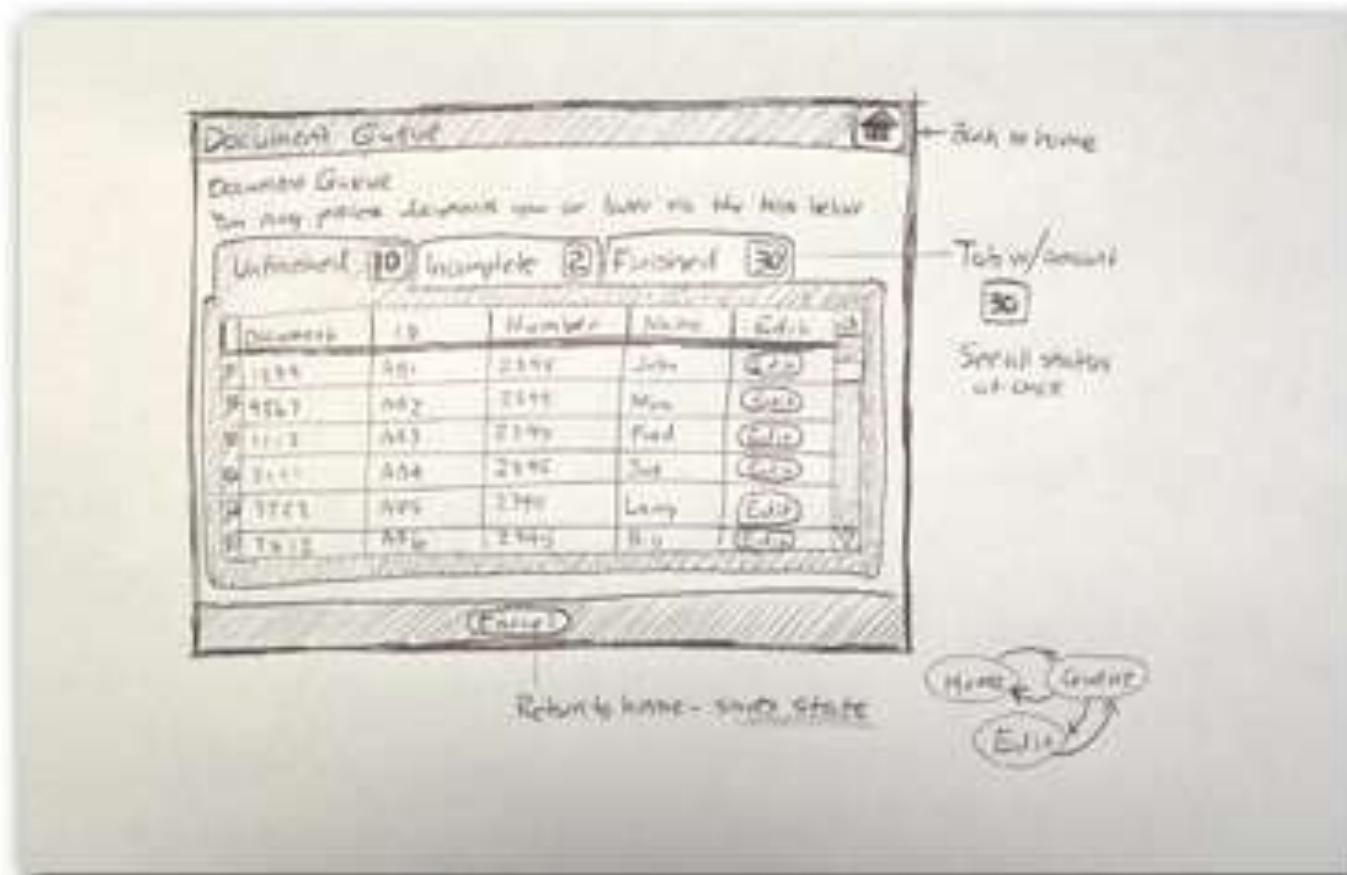
- **Снабдились ноутбуками и переехали к заказчику.**
- **Абстрагировались от изменений в SAP с помощью XSLT.**
- **Loose coupling и сервисный уровень – по максимуму.**
- **Продолжительность спринта – 2 недели.**

- Разделили работы на 3 части (работали втроем над каждым кейсом):
 1. XML\XSLT\BC – Олег Шипинский
 2. Интерфейсы, look and feel – Александр Радзиванович
 3. SQL\SAP Connector, общее руководство и интеграция, тестирование на месте – Игорь Остроухов

- **Итог: спринт завершен в срок; то, что планировалось выполнено.**
- **НО: функционал позже значительно изменился.**

- **Выводы: нужно плотнее общаться с заказчиком, не нужно подключать новых людей в середине проекта**

Работа с Заказиком



**Сколько времени займет дорисовать кнопку?
(Сколько это будет стоить?)**

Общение с Заказчиком

Editing Classes and Assignments for **Whole Test Mentoring** show only planned modules:

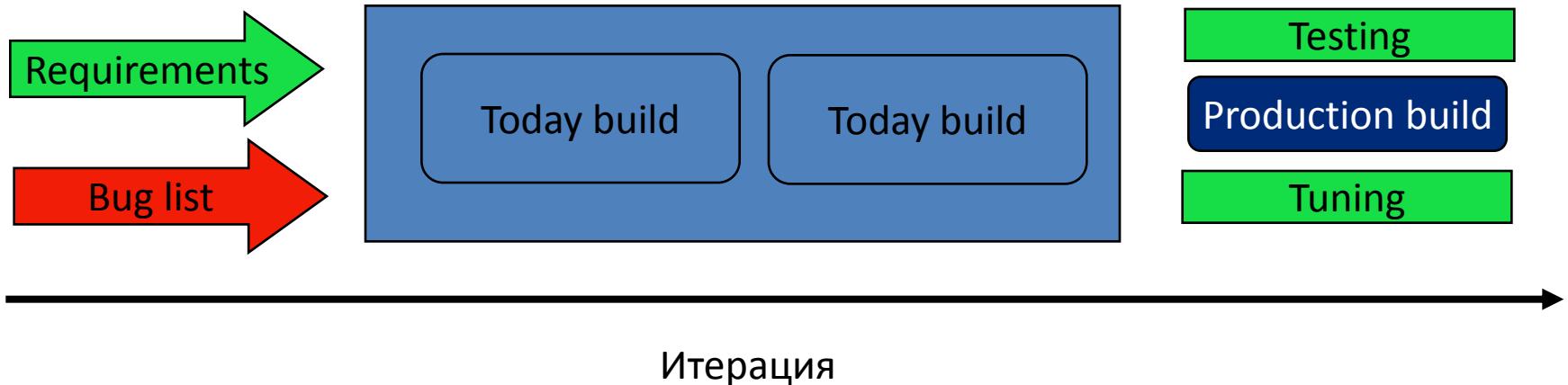
Classes	That appears in Module			Lectures		Assignments	
	Date	Start Time	End Time		Deadline		
Introduction to Project Management	Management for dummies					1.11.2007	<input type="button" value="Add my mentees"/> <input type="button" value="Submit results"/> <input type="button" value="Set scores"/> <input type="button" value="Close"/>
	24.10.2007	10:00	12:00	Add my mentees	Create sandwich		
	25.10.2007	20:00	22:00	Add my mentees			
Introduction to Introduction	Introduction Part 1			Not presented		1.09.2007 (closed)	<input type="button" value="Add my mentees"/> <input type="button" value="Submit results"/> <input type="button" value="Set scores"/> <input type="button" value="Close"/>
Introduction to Requirements Analysis	Requirements Analysis Basics			Not yet scheduled		Not yet scheduled	

Add existing curriculum:

 Add new curriculum:

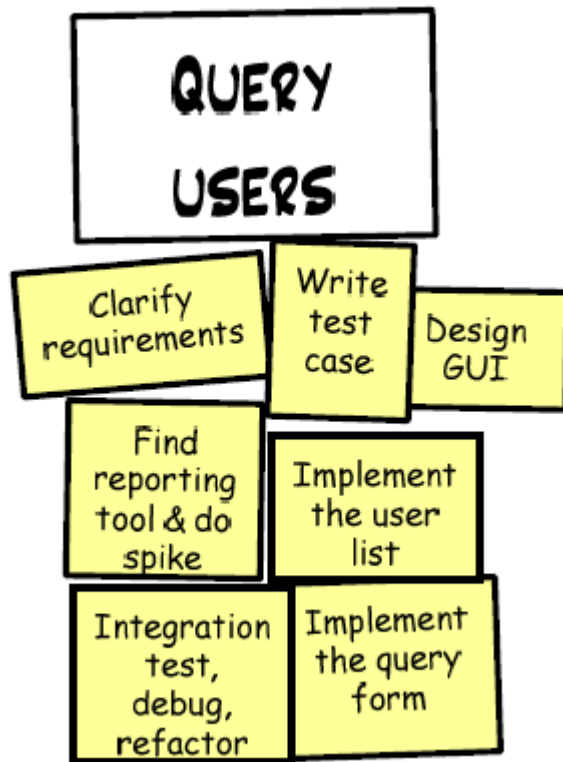
Handwritten notes: "Classes" (with arrow to 'That appears in Module'), "Lectures" (with arrow to 'Lectures' column), "Лекция со обратной связью" (green box), "Create sandwich" (next to 24.10.2007), "Have fun" (next to 1.09.2007), "Add my mentees" (green boxes), "Cancel" and "Save" (circled), "OK" (checked).

Спринт – неделя. Пятница – релиз.



1. Ежеутренние консультации с заказчиком, уточнение требований
2. Установка целей на день и контрольных точек
3. Работа над текущими задачами
4. Интеграция, сборка билда
5. Установка билда на демонстрационный сервер

Что случилось с юзкейсами?



- **Чем меньше задача – тем точнее оценка**
 - Разбивайте большие «хотелки» на меньшие
 - Для каждой «хотелки» расписывайте набор задач
 - Оценивайте каждую задачу
 - Оценивает тот, кто будет делать задачу

Второй спринт – организация бэклогов

- **Столкнулись с новыми проблемами:**
 1. Изменчивость требований – синдром округления глаз, «я такого не говорил...» Вместо 2х строчек – хорошее детальное описание + эскиз.
 2. Поток новых «хотелок» из серии «внесите сразу» - развели бэклоги.
 3. Объективные изменения в SAP и сложности тестирования – сократили итерацию до недели. В конце каждой итерации новая версия с новым куском функциональности.
 4. Халтура. Применили «плантаторский» метод – у каждого есть задача, пока он ее не выполнил, он не уходит.

Ай-ай-ай!

- Мы очень умные. Нам не нужны все agile-практики. Глупости, на которые тратится время, можно выкинуть.

1. Постоянное общение с заказчиком.
2. Постоянное уточнение требований.
3. Постоянное усовершенствование кода.
4. Постоянная переработка функционала.
5. Очень короткие итерации.
6. Коллективное владение кодом.
7. Постоянное тестирование.

- Пренебрежение практиками привело к проблемам.



«Кому они нужны, юнит-тесты?»

1. Тестер не мог тестировать каждую сборку, так как локальный ВС не позволял работать с приложением.
 2. Использовать юнит тестирование и какой-то юнит тестинг фреймворк мы не умели.
 - В этом случае мы совершили чудовищную ошибку – допустили **коллективное владение кодом**.
- **Результат – периодическая потеря кусков функциональности, повторяющиеся ошибки, наведенные ошибки и пр.**

Рефакторинг

- Вкупе с тем, что приложение писалось в страшной спешке наступил очередной переломный момент.
 - Остановились на 4 дня и занялись рефакторингом. В будущем коллективного владения больше не допускали.
 - Тем не менее, среднее количество багов в продакшн билде – 25-30.
 - Вывод: **надо** использовать юнит-тестинг.

Under Time and Pressure

- **Работа шла в максимальном темпе и под довольно большой нагрузкой. В таком случае возникает своя специфика общения и организации рабочего процесса.**
1. Удобные рабочие места. Мощные машины.
 2. Горячее питание. Работать голодным невозможно. Особенно сверхурочно.
 3. Печенье и круассаны. Всегда. Когда жуешь, кровь активней приливает к голове.
 4. Кока-кола. Холодная, запотевшая. Чай – это прекрасно, но летом особенно хорошо рытса в коде, попивая колу или фанту...
 5. Рабочий режим. Если заказчик отпускает программистов не раньше 11 вечера , то пусть они приходят к нему не раньше 11 утра – справедливый обмен.

О субкультурах



Daily Scrum + Tracker

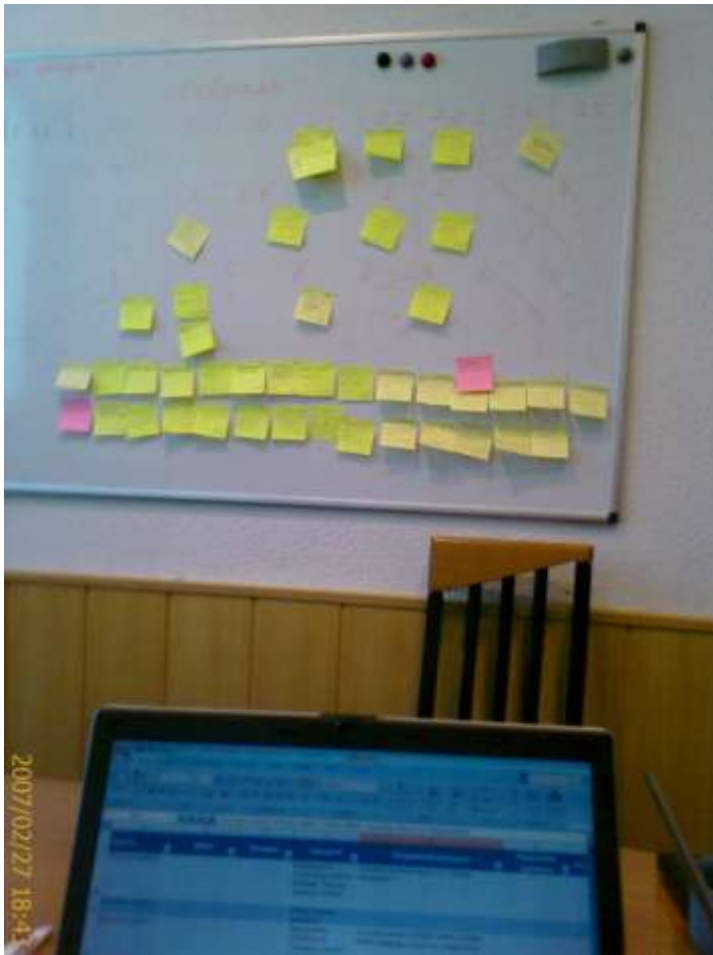


1. Общее собрание раз в неделю – лучше всего в первый рабочий день недели пораньше, чтоб настроить всех на работу в течении недели
2. Утренние разговоры со всеми парнями за первой чашкой кофе вокруг кофеварки (stand up) – важно понять, кто в каком состоянии и настроении, как эффективно развесить нагрузку сегодня
 - i. (Утренние собрания групп – раздача задач и обсуждение открытых проблем)
3. Периодические обсуждения в течении дня – если кто-то уперся в проблему, не дать ему в ней увязнуть (sidebar meetings)
4. Люди всегда должны знать что происходит на проекте, каждый промежуточный успех на проекте должен быть отмечен – на собрании или письмом

«Стоячее совещание»

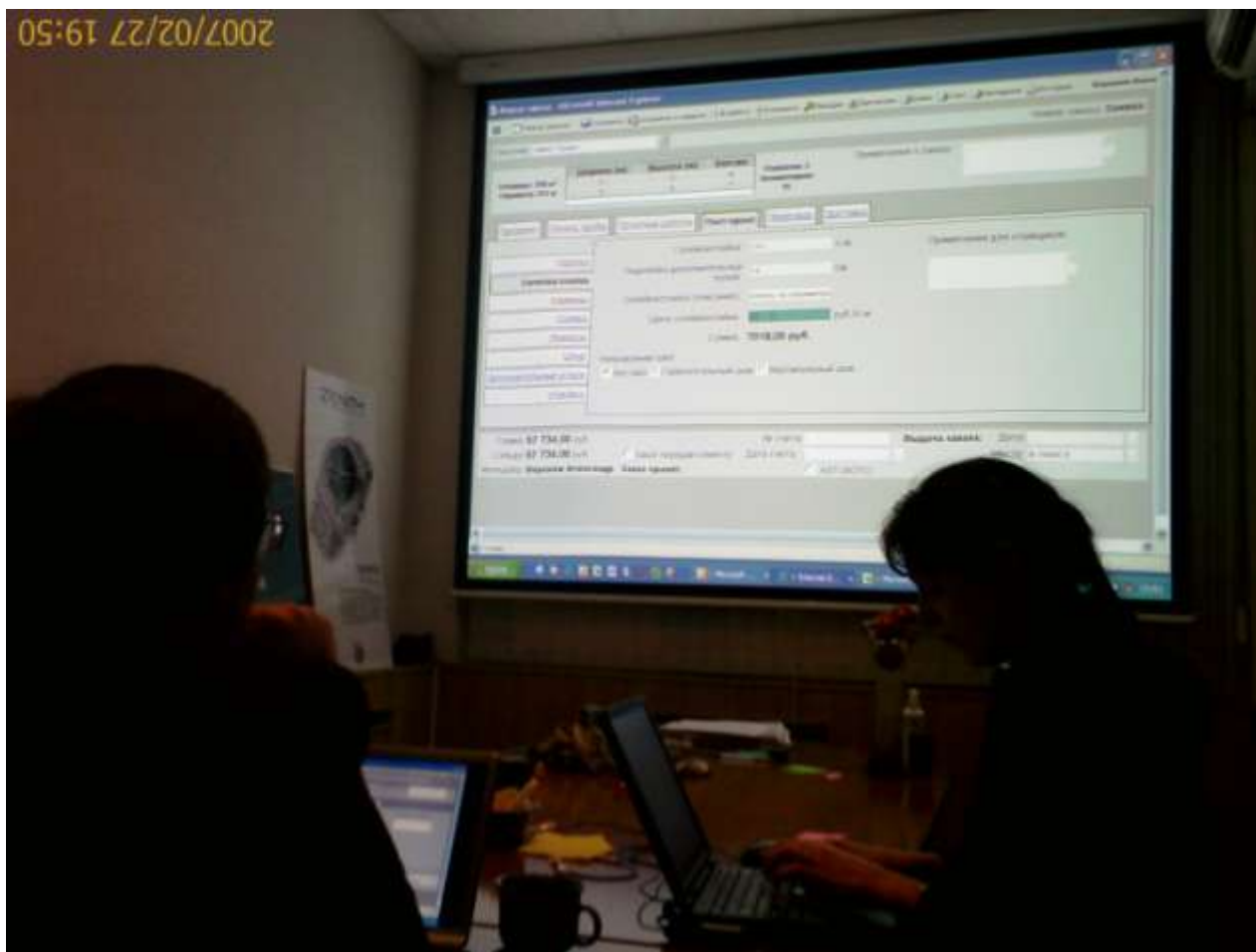


О бэклогах



- Бэклог – это форма записи требований
 - Без бэклога нельзя сделать Заказчика счастливым
- Бэклог – это форма коммуникации
 - Если бэклог непонятен Заказчику – коммуникация не состоялась

О важности демо



Потому что мысль

- **Еще немного о пользе ноутбуков.**
 1. Никаких новых развертываний рабочей среды при переходе с места на место
 2. Необходимый набор программ установленный раз и навсегда
 3. Всегда актуальные данные
 4. Возможность работы где угодно, хоть в метро
 5. Возможность работы в выходные
 6. Многие люди лучше работают дома в выходные
 7. Сознание собственной значимости – великий стимул...
Разработчик, получивший новый гаджет, обязательно продемонстрирует свою значимость поработав дома...

Итого по практикам

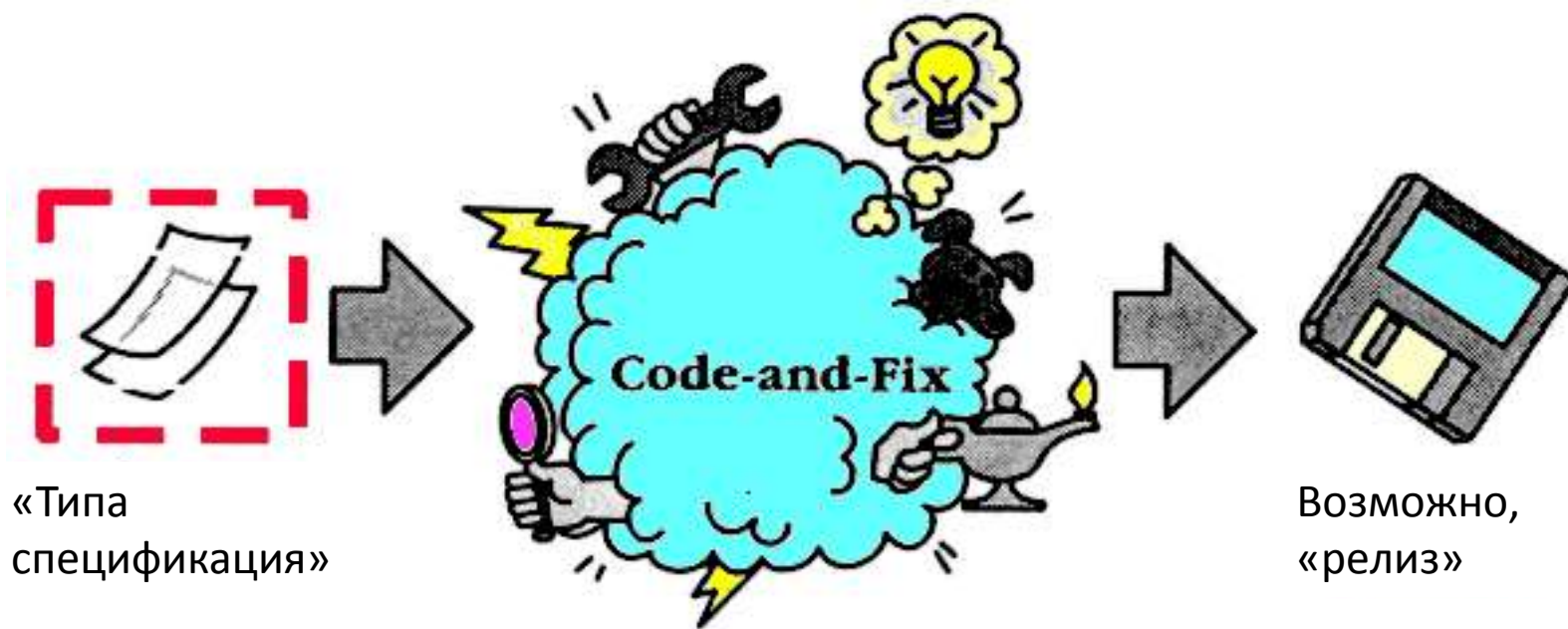
- Заказчиков нужно учить работать правильно
- Команду надо «зажечь» идеей agile
- Короче спринт – больше шансов на успех
- Детализируй до фич, которые можно сделать за день
- Каждому спринту надо задавать heartbeat
- И придерживаться его жестко

Итого



- SCRUM позволяет делать и fixed price-проекты
- Но не надо заикливаться на идеологии – надо помнить, что действительная цель – релиз и прибыль
- Вместе с тем надо очень аккуратно пользоваться практиками
- **«Если ты пренебрег практиками – не говори, что ты делал SCRUM!» (К. Швэйбер)**

Внедряйте все практики



Это не «Agile»!