

XII международная конференция
CEE-SECR / РАЗРАБОТКА ПО

28 - 29 октября, Москва



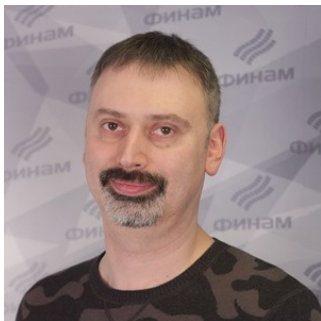
Разработчик всегда прав

как без менеджера общаться с людьми, далёкими от
разработки

Алексей Трошин



Давайте знакомиться



Алексей Трошин

Заместитель директора IT-департамента ФИНАМ

Web-dev с 2002 года, Agile с 2009 (CSPO, CSM)

Project-Product-СТО-..., от 1 до 150 сотрудников

Конференции, семинары, тренинги, книга

Опыт из:



SETUP



...и других:



- (Интроверт) Разработчик может выжить без менеджера вне команды

Кто такой менеджер?

- Пример из должностной инструкции ->
- По сути, это интерфейс
 - Транслирует ожидания заказчика в возможности разработки
 - Транслирует результаты разработки в ожидания заказчика
- И вот его нет...
 - Нам же не нужно делать так? 😊



Один мужик идет вдоль улицы и копает ямы, за ним второй эти ямы закапывает. Прохожий удивляется:

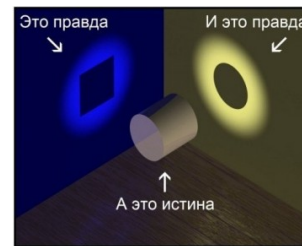
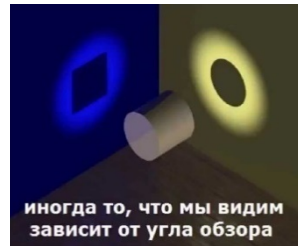
- Вы что делаете, идиоты? Один копает, второй закапывает?
- Я не второй, я третий. Второй не вышел на работу, он должен был деревья сажать

4. Менеджер проектов должен уметь:

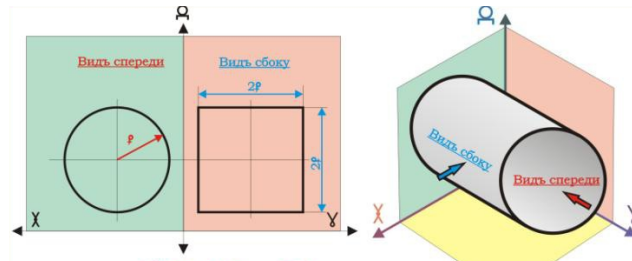
- Планировать работы проекта.
- Подбирать подходящую команду проекта.
- Правильно распределять сотрудников по проекту.
- Формулировать цели и задачи проекта.
- Модерировать совещания.
- Управлять конфликтами.
- Делегировать задач и полномочия сотрудникам.
- Осуществлять контроль над всеми делегированными заданиями.
- Производить расчет расходов проекта.
- Искать пути решения проблем.
- Проводить анализ окружения проекта.
- Делать SWOT-анализ.
- Проводить анализ рисков.
- Строить план структуры проекта.
- Формировать устав проекта.
- Управлять вехами проекта.
- Разрабатывать график работ.
- Проводить переговоры с внутренними и внешними участниками и высшими руководящими органами.
- Ставить цели и задачи по SMART.
- Управлять персоналом.
- Управлять информацией.
- Управлять качеством разрабатываемого продукта.
- Управлять расходами, сроками проекта.
- Управлять проектом по методологиям agile.
- Осуществлять проектный контроллинг.
- Планировать работы.
- Разрабатывать проектные документы, отчеты.
- Использовать инструменты планирования, документирования, контроллинга, управления рисками, качеством, разработкой.

Особенности восприятия

- Пример:



- Всё же просто:



- Требуется обучение языку (или тренинг софт-скиллов)
- Хьюстон, у нас проблема (с) мы хотим быстро и сразу



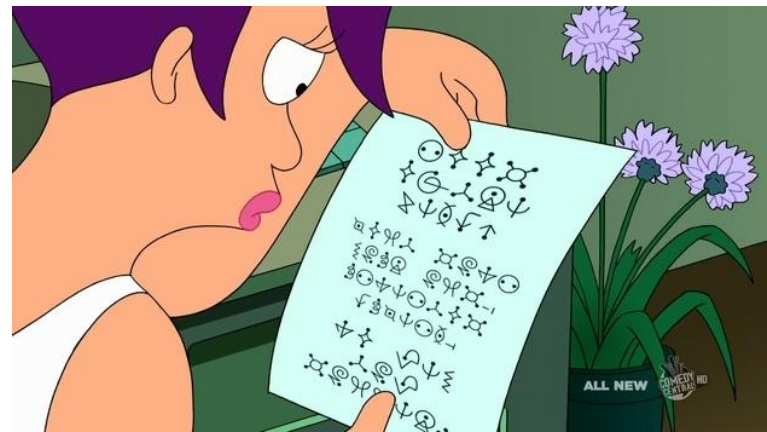
Недопонимание = проблема

- Предание о Вавилонской башне (Библия)
 - Языки перемешались, люди перестали понимать друг друга
 - Проект не завершён



Наши языки

- Наличие общего языка и набора символов не гарантирует понимания!
- Разработчик
 - Технологии, способ реализации
- Менеджер проекта
 - Сроки, качество
 - Мотивация (люди)
- Директор
 - Деньги (доходы, затраты)
 - Клиенты (удовлетворённость, обязательства)
- Бухгалтер (пользователь системы)
 - Отчёт, сверка



Например

- Как общаться с иностранцем?
 - Предположим, он немного знает английский, как и вы 😊
- Короткие предложения
- Простые слова (базовый набор)
- Не стесняться переспрашивать
 - (правильно ли я понял...)
- ...
- Profit!



Что делать?

- Примите тот факт, что вы друг друга не понимаете
 - И вас не понимают
 - Вероятно, вы не в общем контексте
- Старайтесь говорить на языке того, с кем общаетесь
 - Думать как он (не фреймворк, а...)
- Используйте практики, о которых дальше



Перевод на язык бизнеса

- Пересчитываем в деньги/время
 - Страшное настоящее против светлого будущего
- Подкрепляйте аргументами, дайте несколько кейсов
- Правило А4
 - Любой документ не должен быть более одного печатного листа

- Посмотрите доклад
 - <http://morrozmsk.livejournal.com/128782.html>
 - Видео, 30 минут



Пишем письма правильно

- Никто не читает 😊
- Сначала - выводы или варианты решений (закрытый вопрос)
- Возможно, перед этим объяснить контекст (напомнить о чём речь)
- Затем доказательства
- Правило одного экрана
 - Лучше написать мало, чем много
 - Много, скорее всего, точно не прочтут
 - Если чего-то мало – то спросят
 - Вы дадите 100% требуемое уточнение



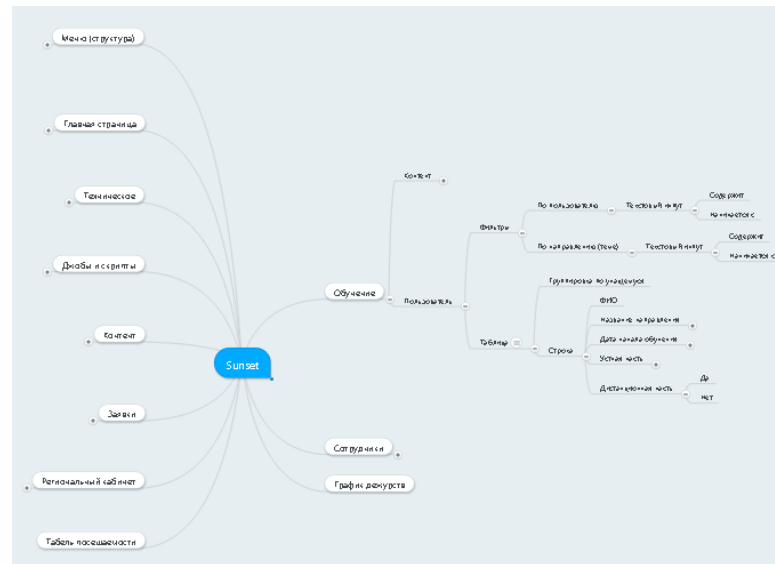
Встреча – формируем ожидания

- Какой результат (с чем выйти)
- Подготовка
- Фидбек по итогам
 - Что решили
 - Кто, Что, Когда
- Ничего не обещать
 - Тем более сроки



Обсуждаем требования

- Никаких букв, никто же не читает 😊
 - Эскизы со спецификациями тоже не смотрят
- Mindmap! Всего 1 экран, большая экономия сил
 - Freemind, XMind
 - Mindmeister.com
- Выделяйте MVP – без чего точно нельзя
 - **Minimal Viable Product**
- План Б – частые встречи с прототипами
 - Люди любят демо
 - Не утомите ;)
- Ничего не обещать 😊



Пишем задачу

- Шаблон для минимального качества
 - Набор обязательных полей
 - Можно и чек-лист, DoD
- Тема (заголовок) – ценность, которая должна получиться в итоге
- TDD – сначала «Критерии приёмки»
 - Как проверить, что задача выполнена
- Затем – описание и «бантики»

User Story содержит несколько информационных блоков, каждый из которых несёт определённую пользу для разных участников разработки:

1. **Название** (для всех)
2. **Суть** (МП, команда)
3. **Информация** (МП, команда)
4. **Приёмка** (МП, заказчик)
5. **Гости** (заказчик)
6. **Тесты** (команда, тестирующий)

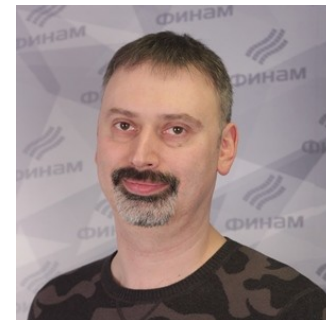
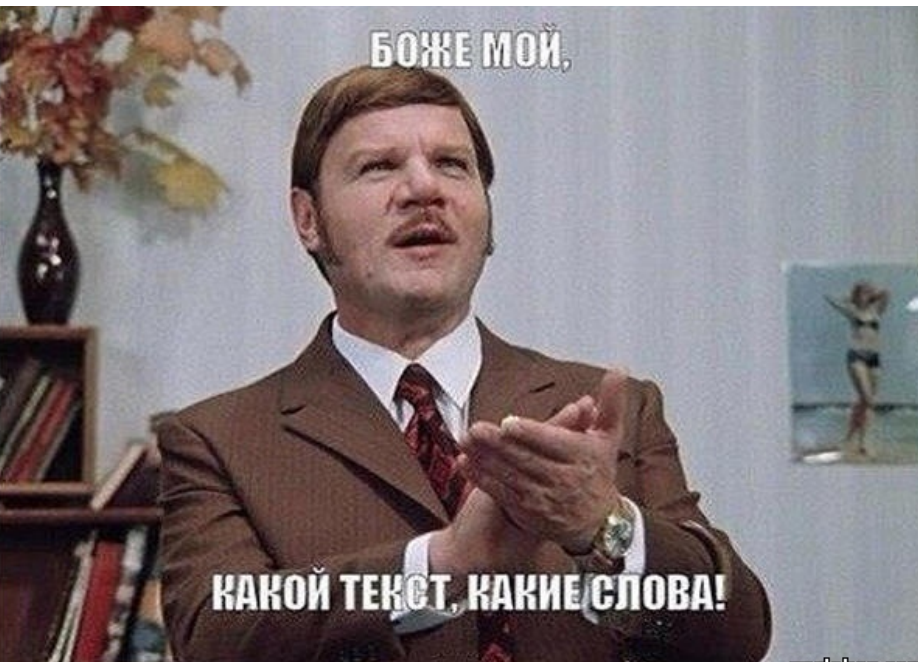
Как бы вывод

- Не нужно умничать – будьте проще (KeepItSimpleStupid)
- Бизнес любит суть и экономию
- Понять и простить 😊
 - Понять – вас не понимают, подстраивайтесь 🐸
 - Простить – использовать эффективные экономные документы
- Ваша цель – дождаться менеджера 😊



Вот и всё 😊

- Поделитесь своими практиками 😊



Алексей Трошин
morrozmsk.livejournal.com

