



#SECONRU



МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ  
РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## Выжить без менеджера: шаблоны правильных коммуникаций

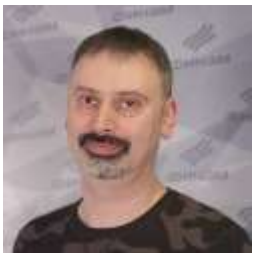
Трошин Алексей

Заместитель СЮ по разработке, ФИНАМ

21-22 АПРЕЛЯ | ПЕНЗА



# 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:



Менеджер уехал...



# 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?

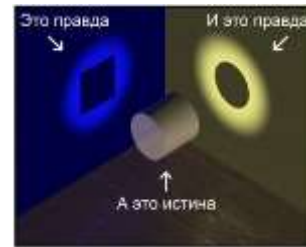
- Я не второй, я третий. Второй не вышел на работу, он должен был деревья сажать

#### 4. Менеджер проектов должен уметь:

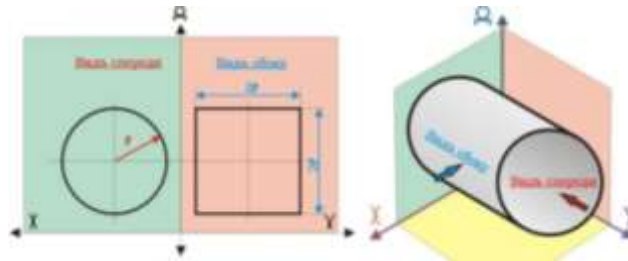
- Планировать работы проекта.
- Подбирать подходящую команду проекта.
- Правильно распределять сотрудников по проекту.
- Формулировать цели и задачи проекта.
- Модерировать совещания.
- Управлять конфликтами.
- Делегировать задач и полномочия сотрудникам.
- Осуществлять контроль над всеми делегированными 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 сроками проекта.
- Разрабатывать графики 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 разработки.

# Особенности восприятия

- Пример:



- Всё же просто:



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



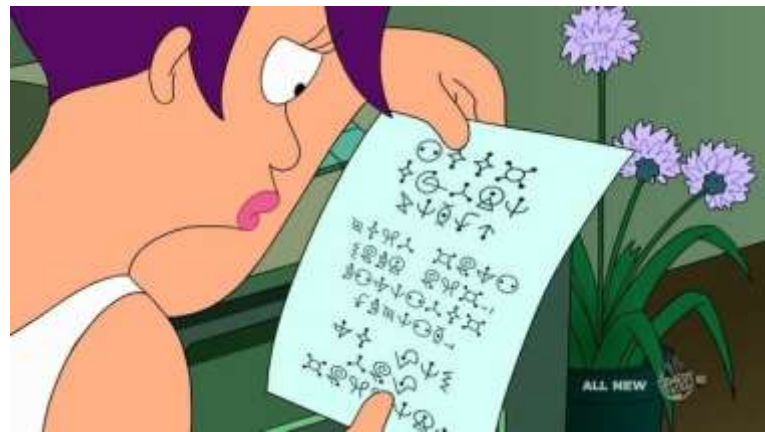
# Недопонимание = проблема

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



# Наши языки

- Наличие общего языка и набора букв 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!





Они чужие, среди нас 😊



# 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

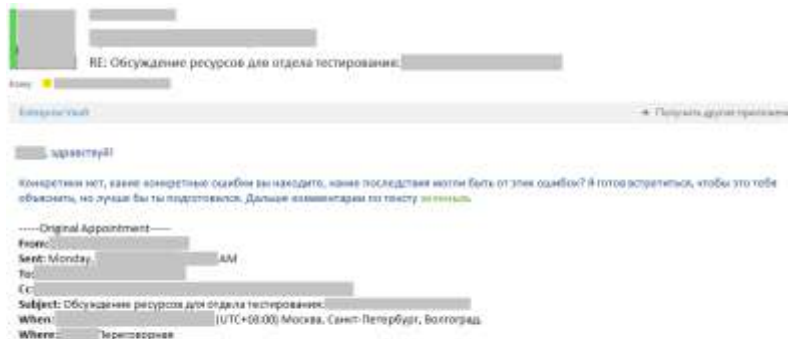


# Перевод на язык бизнеса

- Пересчитываем в деньги/время
  - Страхное настоящее против светлого будущего
- Подкрепляйте аргументами, дайте несколько кейсов
- Правило А4
  - Любой документ должен быть **не более** одного печатного листа
  
- Посмотрите доклад
  - <http://morrozmsk.livejournal.com/128782.html>
  - Видео, 30 минут



# Пишем письма правильно



Для полноценного обеспечения качества на проекте [redacted] и [redacted], необходимые ресурсы:

- Младший менеджер по тестированию (50.000-80.000)
- Младший инженер по тестированию (1-40.000)

Детально, что конкретно входит в «обеспечение качества», я написал в самом письме, с указанием пользы от каждого типа проводимых работ. Если кратко, то это весь путь от тестирования требований до нагрузочного тестирования.

В рамках проекта QM данные сотрудники также будут привлечены к другим работам в целях обмена опытом и взаимовыгодности. А так же [redacted] по [redacted] фокусирован.

#### Обоснование:

- Это простой или сложный ввод в финансовый поток, вероятно больше, чем стоит ресурс тестирования, который уменьшит сроки проекта. *Ввод нет конкретика, аргументов - это не доказано*
- На текущий момент мы потрбовали выделить 2 человека на 20% с проекта [redacted] и стали находить до 5 ошибок на (незначительной) релиз, не найденные другими сотрудниками. *Статья вводит в заблуждение, что их можно разгрузить от рутинной детальной проверки функционала (регрес). Если выверится на релиз, а можно и релиз вообще безболезнен?*

- Работоспособность и функциональность сайта напрямую ведет к финансовым рискам для компании (как ст. закон 37 лет без тестирования, это актриса? Это страшно и требует тестирования?)
- С текущим ресурсом [redacted] (20% и в ущерб [redacted]), уже невозможно увеличить количество найденных дефектов на ранней стадии, что удваивает их исправление, но большая часть задач пока не тестируется. *Ввод нет конкретика*
- На исследование дефекта тратится меньше времени разработчика, что экономит их время, более дорогое по сравнению с Тестировщиком. *Ввод, нет конкретики. По какому по времени пишется. Если не значит что разработка дороже - я тебе повторю что это не так*
- Добавление тестировщика позволит выпускать более контролируемые и продаваемые релизы [redacted] с чем у нас возникли сложности, т.е. у разработчиков меньше времени будет уходить на локализацию и поиск дефектов, они больше времени смогут уделять новую функционал. *Статья вводит, много конкретики, обоснование адекватно описано будущим*
- Сайт будет регулярно тестироваться на работоспособность ранее выявленного функционала, интеграций с другими системами (то сейчас производится с низкой интенсивностью и несет риски) *Нет конкретики - что мы, немедленно сразу прекратить, а куда?*
- Оценка стабильности системы - проведение регулярного нагрузочного тестирования позволяет регулярно собирать информацию о текущих рисках и принимать превентивные меры для реагирования на них, до возникновения инцидентов (сразу тапалетелл) *Статья вводит в заблуждение без какой-либо конкретики*

#### Выводы:

- привлечение не очень позитивно сказывается на их эффективности, т.е. на текущий момент каждый тестировщик уже работает минимум с двумя проектами. Ну и где видно что они работают на нескольких проектах - таблица бы
- фокусировка на проекте для получения максимальной эффективности. Так фокусировка или не эффективна? Мысли проекта требует полного погружения, частично тестировать QM, Автообслуживание, Ресур, Ресур, ли просто невозможно, они требуют погружения большого объема информации и временных затрат, что конфликтует с распределением на 2Я проект.

#### Подробнее про обеспечение качества.

- Обеспечение качества на проектах вводит:
  - Систематизацией вояк и контроль дедлайнов на проектах
  - Контроль выпускаемых релизов (полнота, качество, оценка возможных рисков) *Не понимаю как это соотносится с качеством*
  - Тестирование задач до релиза: удешевление стоимости исправления багов и за более раннего обнаружения
  - Проведение приемочного тестирования до релиза: уменьшение риска неудачного релиза на бой
  - Проведение нагрузочного тестирования
  - Проведение регрессионного тестирования: уменьшение рисков поломки старого функционала

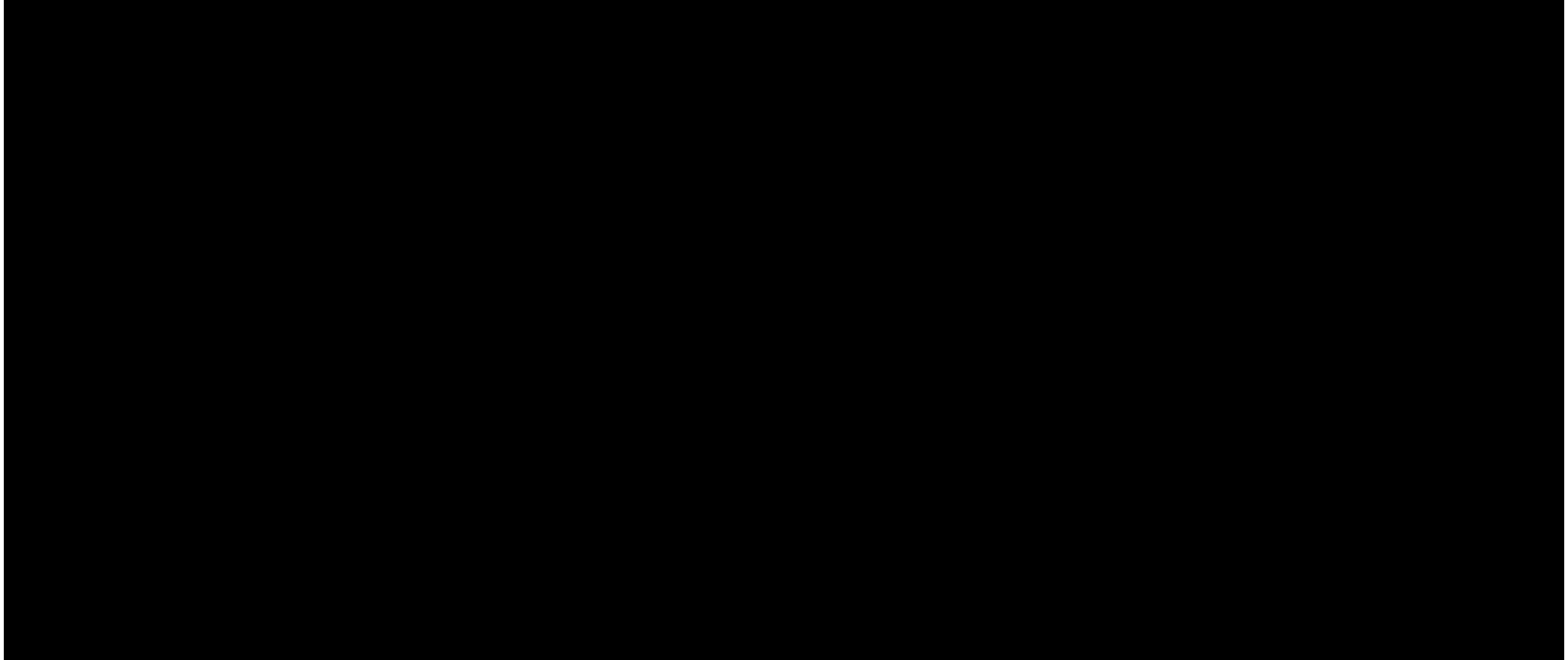
На текущий момент на обоих проектах, выделен ресурс на 20%, в рамках с других проектов, для оценки целесообразности тестирования, тестирование не очень успешное, доводит до конца проекты, согласно с необходимостью проводить регулярное тестирование. Это за тестирование? Финансы не ограничены? Как посмотреть на результаты тестирования? Это эти данные: регулярные? Чем конкретно они отличаются?

# 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



# Идеальное принятие решения 😊



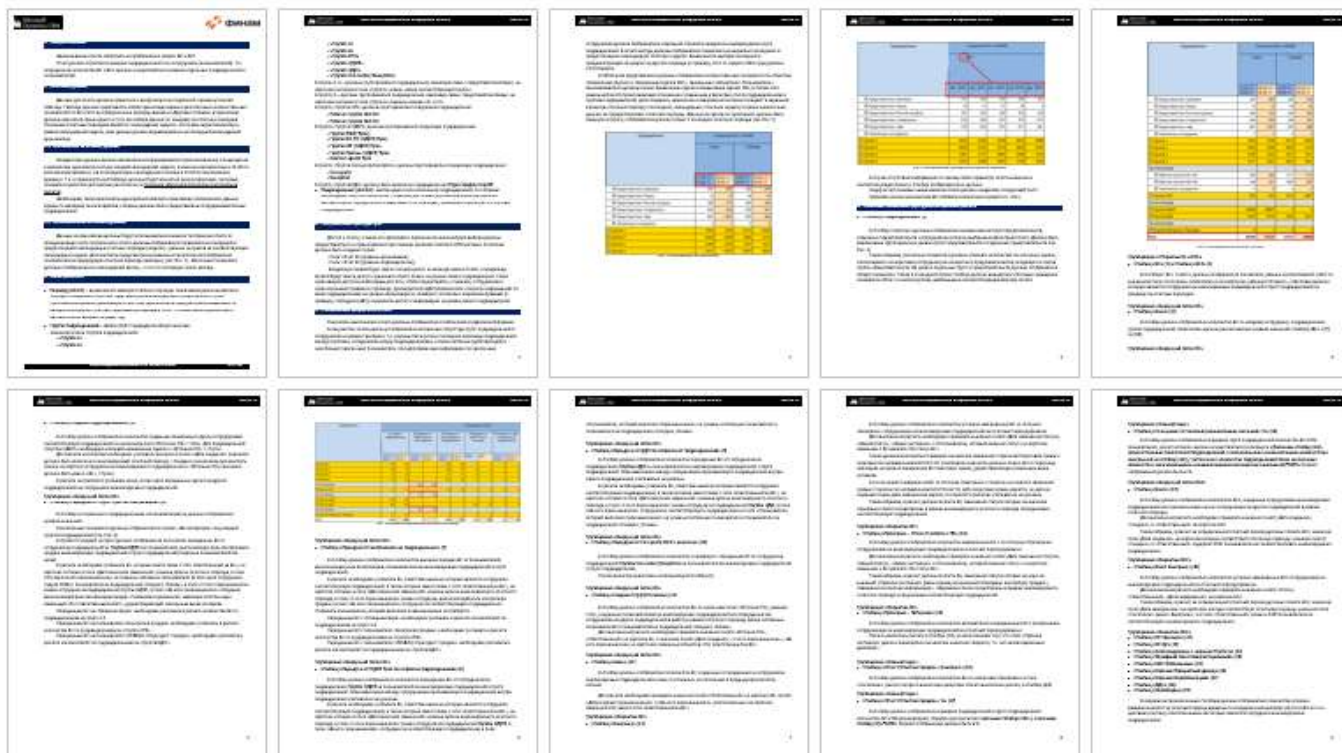
# Встреча – формируем ожидания

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

Даже самую простую задачу можно сделать невыполнимой, если провести достаточное количество совещаний.



# Обсуждаем требования



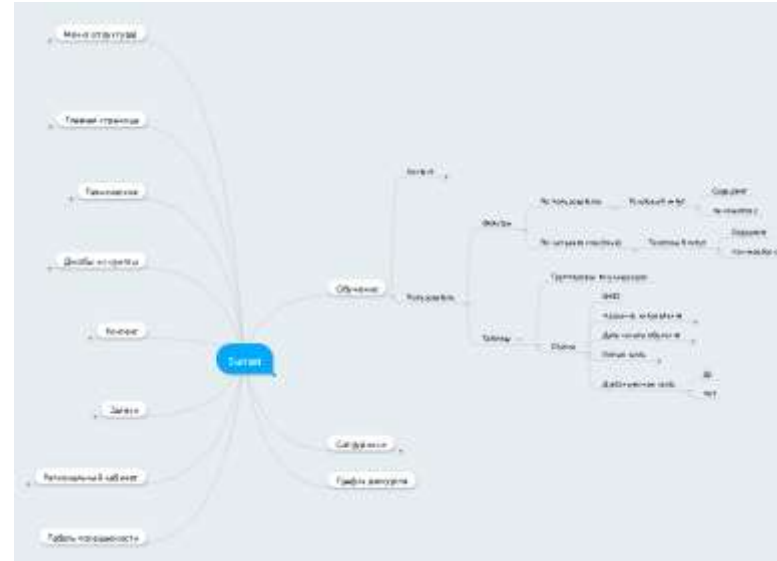
The image displays ten slides from a presentation, arranged in two rows of five. Each slide contains text, tables, or charts. The tables and charts use a color-coded system: blue for headers, yellow for data, and orange for specific categories or highlights.

- Slide 1 (top-left):** Contains text and a small table with 2 columns and 4 rows.
- Slide 2 (top-second):** Contains text and a table with 2 columns and 4 rows.
- Slide 3 (top-third):** Contains text and a table with 2 columns and 4 rows.
- Slide 4 (top-fourth):** Contains text and a table with 2 columns and 4 rows.
- Slide 5 (top-fifth):** Contains text and a table with 2 columns and 4 rows.
- Slide 6 (bottom-left):** Contains text and a table with 2 columns and 4 rows.
- Slide 7 (bottom-second):** Contains text and a table with 2 columns and 4 rows.
- Slide 8 (bottom-third):** Contains text and a table with 2 columns and 4 rows.
- Slide 9 (bottom-fourth):** Contains text and a table with 2 columns and 4 rows.
- Slide 10 (bottom-fifth):** Contains text and a table with 2 columns and 4 rows.



# 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 (заголовков) – ценность, которая должна получиться в итоге
- TDD – сначала «Критерии приёмки»
  - Как проверить, что задача выполнена
- Затем – описание и «бантики»

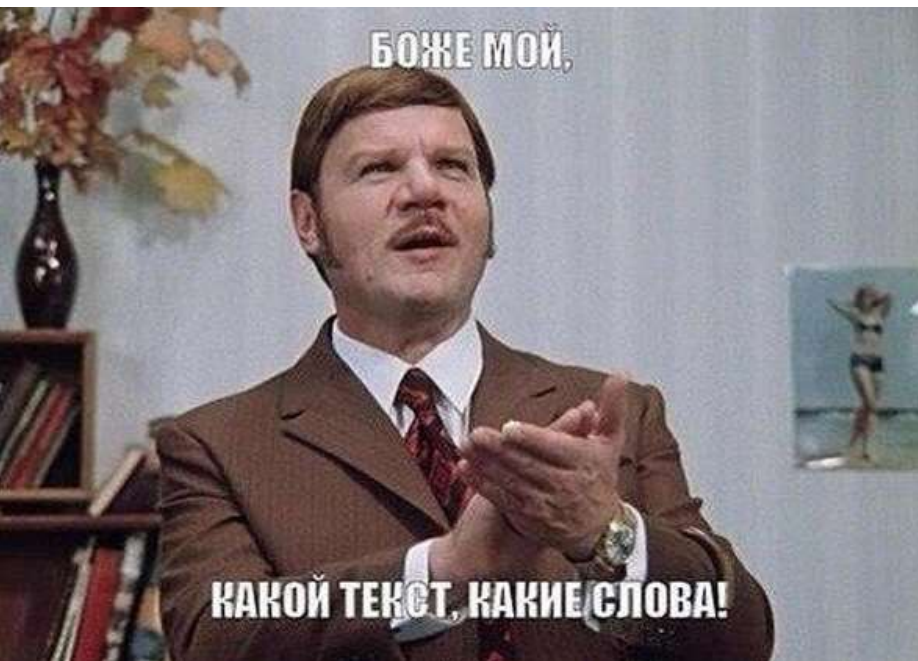
User Story содержит несколько информационных блоков, каждый из которых несёт определённую пользу для разных участников разработки:

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 – пишите меньше
- Понять и простить 😊
  - Вас не понимают, подстраивайтесь 
  - Не изобретайте, используйте шаблонные документы
- Не обещайте! Дождитесь менеджера 😊





Алексей Трошин  
[morrozmsk.livejournal.com](http://morrozmsk.livejournal.com)

