

# Круглый Стол

## «Какие аналитики нужны?»

Эффективное разделение ролей и  
обязанностей

# Описание проблемы



# Человек оркестр



Очень  
эффективно

- Сам снимает требования
- Сам проектирует
- Сам программирует
- Сам тестирует
- Сам внедряет

Не  
работает в  
больших  
проектах

# Классическая схема взаимодействия

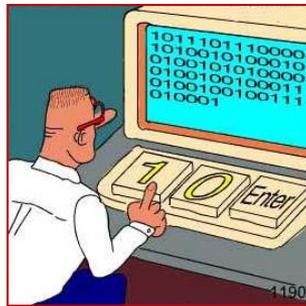
q/a

dev

с/а

б/а

Заказчик



Работающий  
продукт

ТЗ

формализованные  
требования

неформализованные  
требования

# А если народу совсем много то:



Главный Q/A



Tech Lead



Архитектор



# А еще техсуппорт и внедрение



# Бизнес аналитик



- ✓ Эксперт в предметной области
- ✓ Говорит с заказчиком на одном языке
- ✓ Собирает требования с Заказчика ( И согласует их с ним)

- ✓ В состоянии предоставить знания в структурированном и формализованном виде
- ✓ В состоянии отличить важное от не важного
- ✓ В состоянии описать Use-case
- ✓ В состоянии «Проверифицировать» модель системного аналитика

# Системный аналитик



- ✓ Держит общую концепцию в голове
- ✓ Систематизирует знания
- ✓ Борется со сложностью
- ✓ Стыкует бизнес и ИТ

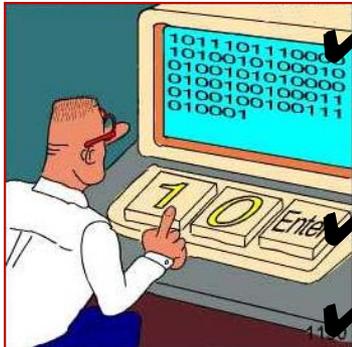
## ✓ Строит модели (Проектирует)

- Проверяет на полноту и не противоречивость
- Придумывает «Абстракции» (сознательно игнорирует маловажные детали)
- Делит на слабосвязанные части
- Осознает и обозначает границы модели

## ✓ Объясняет модели разработчикам и б/а

## ✓ Пишет задание на разработку

# Разработчик



- ✓ Продумывает «технические детали» реализации
- ✓ Проверяет модели с/а на реализуемость
- ✓ Реализует (отливает в железе)

# Quality assurance



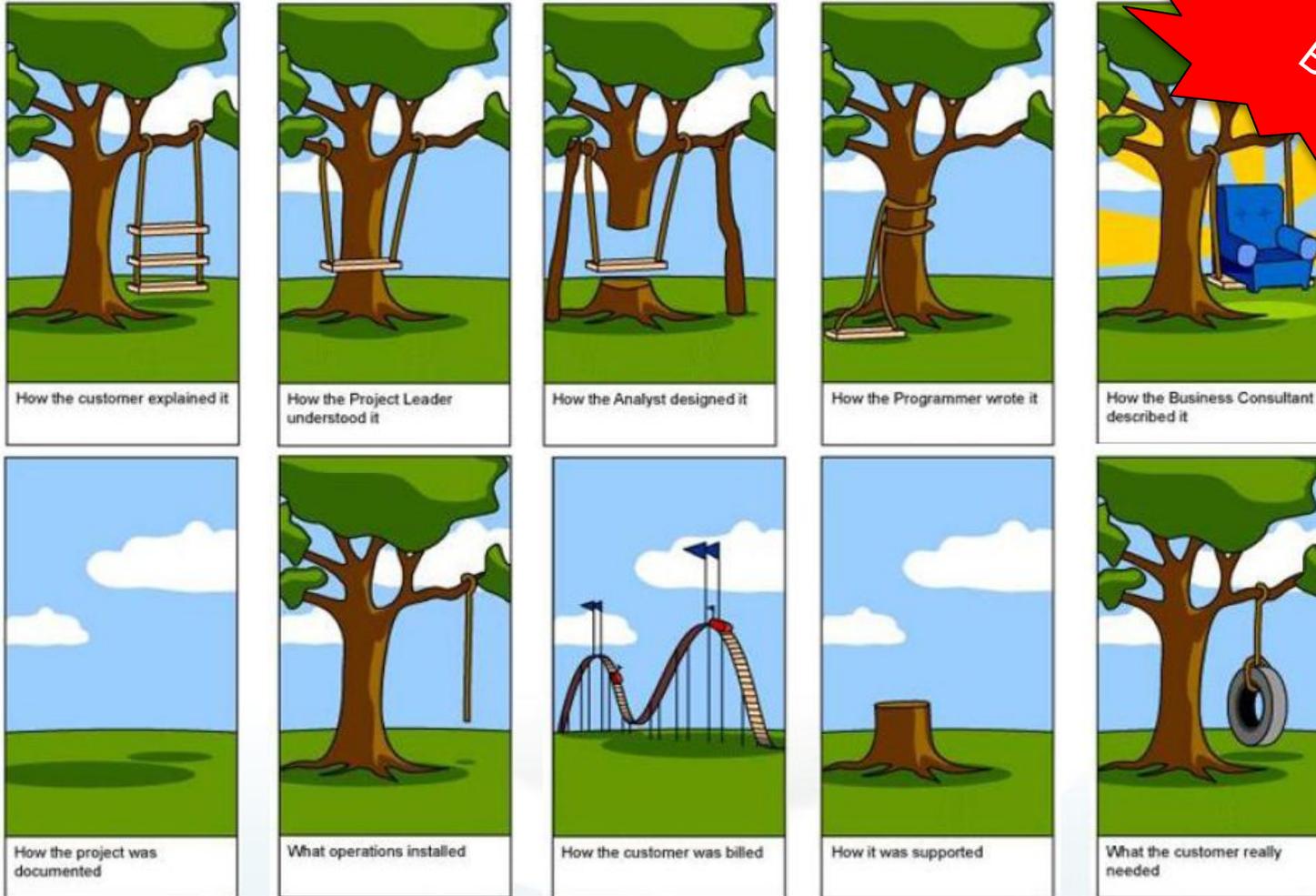
✓ Проверяет реализованное ПО :

- соответствие модели
- удобство использования
- возможность реализовать описанные

✓ Ищет технические ошибки

# Собственно проблемы

# Потеря контекста на звеньях передачи



# Бизнес аналитик



Птица «Говорун»



Не строит модели:

- ✓ Не может проверить полноту требований
- ✓ Не может проверить их непротиворечивость
- ✓ Не может ответственно обсуждать с заказчиком варианты реализации
- ✓ Челночные переговоры (Заказчик <-> б/а <-> С/а )

Ни за что реально не отвечает.

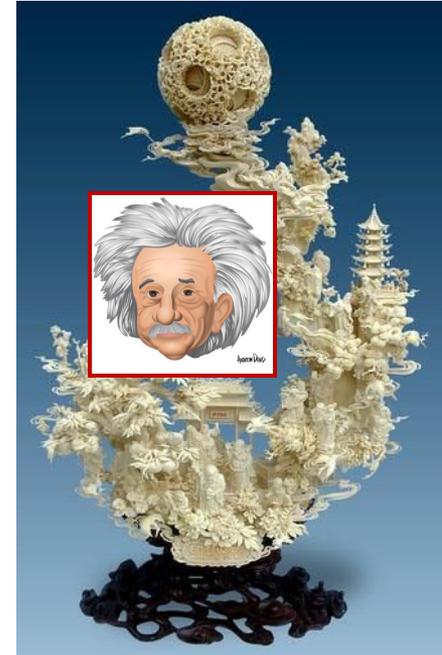
Не пользуется авторитетом у заказчика

Не пользуется авторитетом у разработчиков

# Системный аналитик

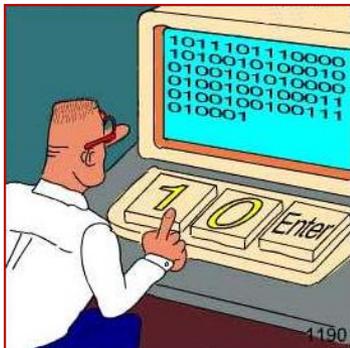


Мудрец в башне  
из «Слоновой  
Кости»



- Не общается с заказчиком (пользователем):
- Не знает деталей реализации
- Оторван от земли.. (Чистые абстракции)

# Разработчик



- Изолирован от заказчика:
  - Больше всего влияет на результат  
(Реализовано будет только то, что понял программист)
- Ни за что не отвечает:
- Ошибок нет? ТЗ соответствует? ко мне какие вопросы?

# Quality assurance



Мальчик для  
битья



- Отвечает за все( «Последний бастион качества», Все шишки в начале – q/a «Как вы это пропустили?» )
- Последний в цепочке получения информации...
- Ни на что не влияет (Никаких решений не принимает)

# Классические способы борьбы



Подробные  
спецификации

И все проблемы  
водопада

## Обсуждение:

- Какая схема работает у вас в компании?
  - Какие проекты делает компания ?
  - Выделена ли у вас роль Аналитика?
  - Есть ли у вас разные роли для Аналитиков?
  - Как аналитики взаимодействуют с заказчиком?, разработчиками? q/a? Поддержкой? Техсуппортом? внедренцами?
  - Вы довольны? Какие есть проблемы?
- Какую схему вы считаете более эффективное?

## Опыт CUSTIS® (Торговые Сети)

Особенности:

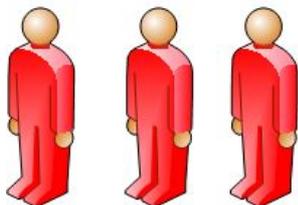
- ✓ заказная разработка
- ✓ длительная (несколько лет) работа команды над одним продуктом
- ✓ б/а – во многом на стороне клиента
- ✓ а так же на клиенте:
  - Техническая поддержка (Первая и вторая линия)
  - Обучение пользователей

# Роли в команде – Вариант 1 (Рук – Tech Lead)

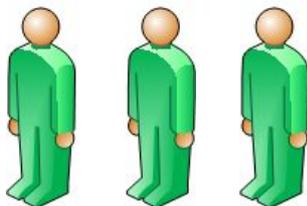
Руководитель



Разработчики



Инженеры  
Аналитики

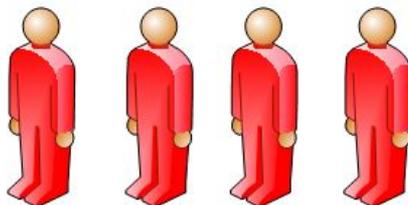


# Роли в команде – Вариант 2 (Рук – главный Q/A)

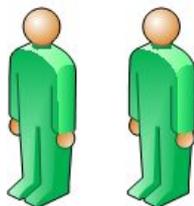
Руководитель



Разработчики



Инженеры  
Аналитики



# Общий контекст

- ✓ Небольшие команды (5-9 человек)
- ✓ Одна замкнутая предметная область
- ✓ Все члены команды погружены в предметную область  
(насколько возможно)
- ✓ «Экскурсии» и «Рекогносцировки» на местах реального использования (для всех членов команды)
- ✓ Единое информационное пространство с заказчиком (wiki, bugzilla)

# Минимизация цепочки передачи информации

Не везде соответствует жизни. Но на уровне базовых принципов - верно

- ✓ Аналитик обязательно совмещен
  - С разработчиком (знает детали реализации)
  - С инженером (общается с пользователем, знает реальные случаи использования, ведет техническую поддержку 3го уровня)
- ✓ Разработчики тоже пишут постановки и участвуют в переговорах с заказчиком (а также во внедрениях).
- ✓ И Разработчики и инженеры участвуют в принятии принципиальных решений
- ✓ Инженеры участвуют во внедрении/ обучении пользователей (по крайней мере первый раз)

## Общая ответственность перед пользователем

- ✓ Все члены команды знают кто их заказчик и пользователь (и хотя бы раз его видели).
- ✓ Критерий успеха – работающее ПО у клиента
- ✓ Заказчик приезжает на демонстрацию в команду. (каждый член команды САМ показывает заказчику – что он сделал)

Не везде соответствует  
жизни. Но на уровне  
базовых принципов -  
верно

# Исключения – выделенный системный аналитик

- ✓ Исторически еще есть...
- ✓ Скорее плохая практика чем хорошая...

# Исключения – выделенный бизнес аналитик

✓ Бывает необходим для очень запутанных предметных областей

Например: для Биллинга ЖКХ - нужно знать всю законодательную базу что бы общаться с заказчиком