

ПЛАНИРОВАНИЕ И ЗАПУСК ПРОЕКТА ПРИ НУЛЕВОЙ ВИДИМОСТИ – ПРАКТИЧЕСКИЙ ОПЫТ

Михаил Ганчиков, руководитель проекта
First Line Software 2011

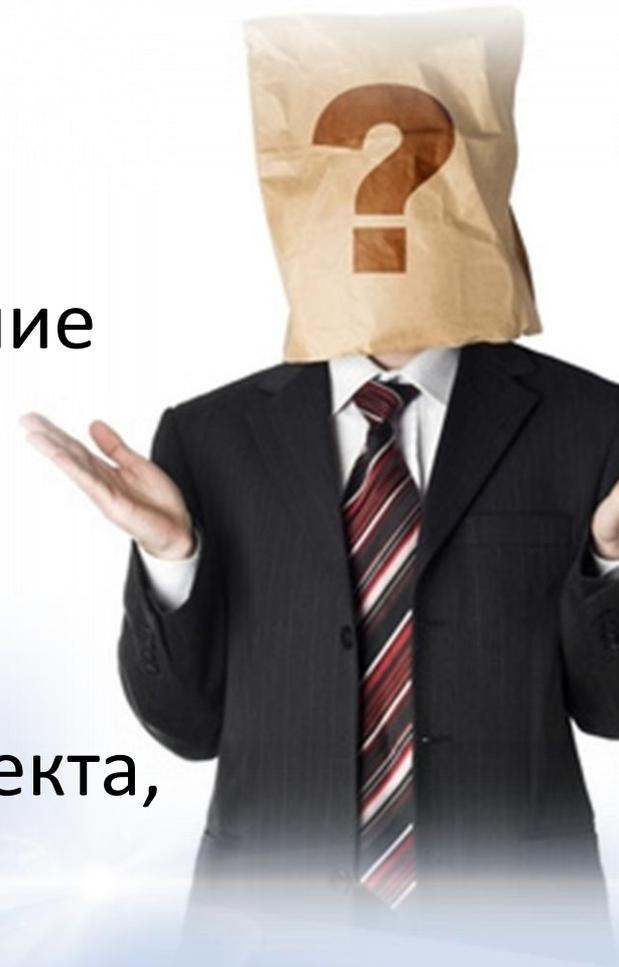
Think Results.

О ЧЁМ ЭТА ПРЕЗЕНТАЦИЯ?

- **Нулевая видимость** – это когда о проекте неизвестно ничего, кроме сроков
 - ...и требования выполнить все требования
- Есть масса способов избежать такой ситуации
 - но мы не будем их обсуждать
- Что делать, если не удалось избежать?
 - управлять бюджетом

СОДЕРЖАНИЕ

- 1. Оценка проекта, планирование бюджета и роста команды
- 2. Работа с требованиями и отслеживание изменений
- 3. Контроль за состоянием проекта, внесение корректировок
- 4. Недостатки подхода
- 5. Полученные результаты и выводы



1. АНАЛИЗ И ОЦЕНКА ТРЕБОВАНИЙ

- Трата времени на сбор требований может не дать желаемого результата
- Для старта проекта достаточно понимать функционал на концептуальном уровне
 - и иметь несколько детализированных историй
- Оценка историй относительно друг друга
 - позволяет абстрагироваться от человеко-дней

ПРОБЛЕМЫ И РЕШЕНИЯ

- Пробелы в требованиях
 - заполняются предположениями команды
- Неясность технической реализации
 - определяет степень риска для каждой истории
 - риски влияют на длительность проекта



ОЦЕНКА ПРОЕКТА

- Общий объём работ:

$$N_{Person\ Days} = \sum_{St=0}^M \text{Story Points}_{St} * \text{Risk}_{St}$$

, где St – история, M – общее кол-во историй, $Risk$ – коэффициент риска

- Проектная норма риска (SP в день):

$$\text{Ratio} = \frac{\sum \text{Story Points}}{N_{Person\ Days}}$$



ПЛАНИРОВАНИЕ РОСТА КОМАНДЫ

- Наличие полной команды в начале проекта
 - может замедлить ход и демотивировать людей
- Строительство команды
 - лучше вести итеративно, наращивая ядро вокруг lead'ов и senior'ов



ПРОИЗВОДИТЕЛЬНОСТЬ КОМАНДЫ

- Лишь часть команды
 - влияет на её производительность
- При росте числа людей
 - сокращается производительность ядра
- Новые члены команды
 - не могут работать с той же скоростью, что и все остальные

ПЛАН ВЫПОЛНЕНИЯ РАБОТ

- Плановая норма на sprint (Target burn-up):

$$S_{Story\ Points} = N_{Developers} * Ratio$$

, где N – “производящая” часть команды



2. РАБОТА С ТРЕБОВАНИЯМИ

■ Technical Product Owner

- ищет простые решения
- формулирует технические задачи для команды
- анализирует результаты приёмки и тестирования заказчиком
- определяет изменения (change requests) функционала



РАБОТА С ТРЕБОВАНИЯМИ - 2

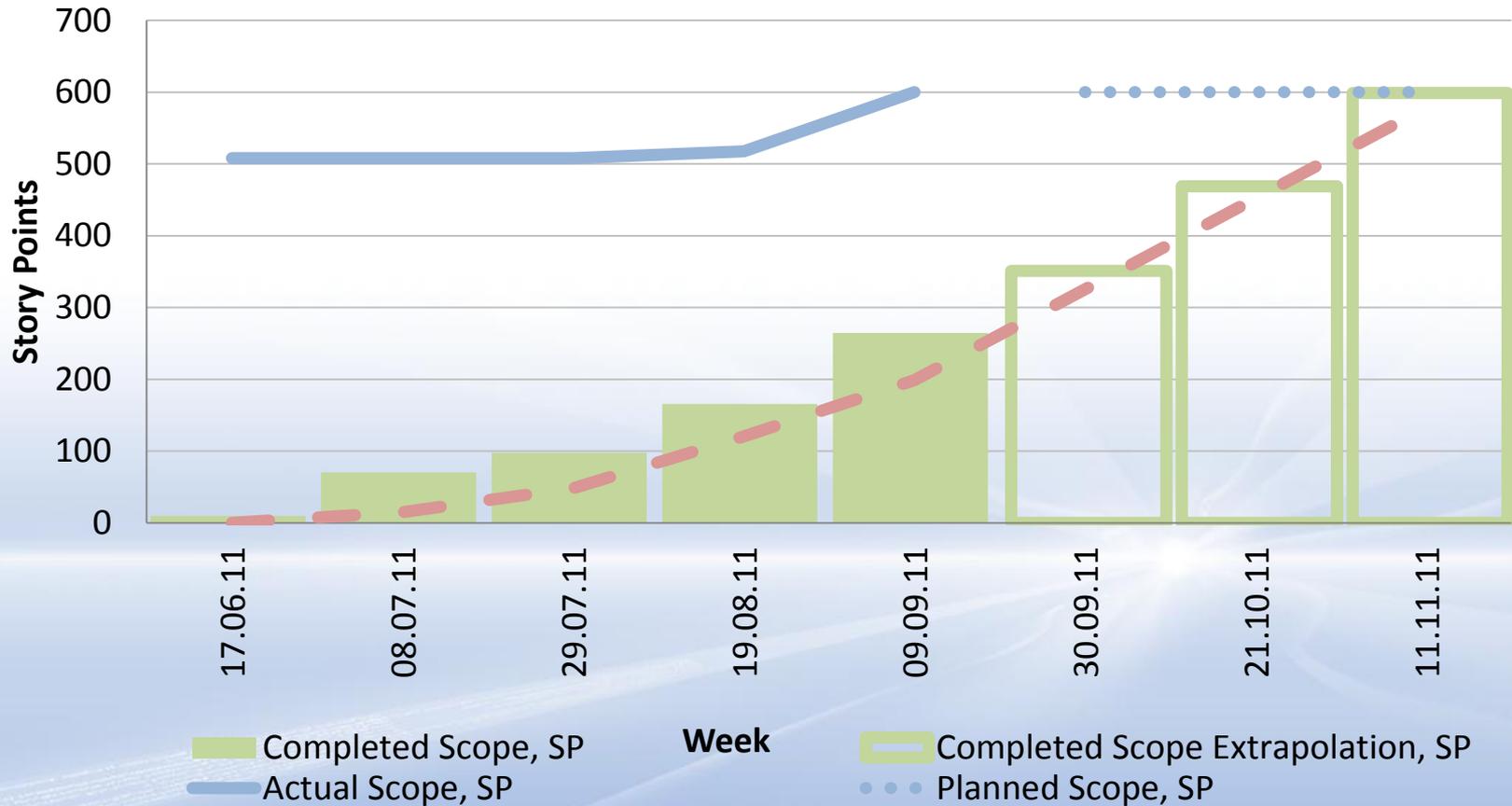
- Команда может помочь ТРО в поиске технического решения
 - выполнения исследовательские задачи
- Product Owner должен следить за приоритезацией историй в backlog'e
 - и помнить о фиксированном количестве story points, которые может сделать команда

3. КОНТРОЛЬ СОСТОЯНИЯ ПРОЕКТА

- Анализ состояния проекта по 3 параметрам (story points):
 - **плановая** норма на спринт (Target burn-up)
 - **фактический** объём выполненных историй (Completed scope)
 - **общий** объём product backlog'a (Actual scope)



КОНТРОЛЬ СОСТОЯНИЯ ПРОЕКТА - 2



ЧТО ДАЛЬШЕ?

- Почему нужна переоценка бэклога в середине проекта?
 - появление новых историй – change request'ов
 - детализация требований для существующих историй может повлиять на их оценку
 - полученные знания о предметной области позволяют сделать оценку более точной
- В случае роста бэклога – эскалировать руководству

4. НЕДОСТАТКИ ПОДХОДА

- Снижение производительности
- Снижение качества продукта

Ratio (bugs/SP)



СПОСОБЫ БОРЬБЫ ЗА КАЧЕСТВО

- TDD (test driven development)
- Автоматизированное регресс-тестирование
- Ассертансе тестирование на стороне заказчика
 - составление сценариев на основе ассертансе-тестов



5. ВЫВОДЫ И РЕЗУЛЬТАТЫ

- **Закладка рисков в оценку**
залог того, что проект не останется без запаса по времени
- **Факторы роста команды, нагрузки на ключевых игроков**
должны учитываться при расчете объема работ
- **Работа с требованиями**
должна продолжаться до конца проекта
- **Мониторинг плановых и фактических показателей**
позволяет не упустить момент, когда нужно просить заказчика остановиться и пересмотреть требования

СПАСИБО ЗА ВНИМАНИЕ

- Вопросы?
- mganchikov@1st-sw.com