

От Shareware к профессиональной разработке софта. Опыт одной маленькой компании.

Андрей Малышев

ISDEF 2018 Spring

Ростов на Дону, апрель 2018

Немного истории

- Одни из первых в российской shareware индустрии
- Первый продукт выпущен в середине 90х (золотые времена)
- Изначально продукты создавались «для себя»

Как все просто было в эпоху shareware

- Каждый разработчик писал свой продукт
- Прямое управление: директор -> разработчик
- Можно было разрешить писать на чем угодно
- Акционеры фирмы занимались абсолютно всем, включая программирование

Первые проблемы

- Незаменимые сотрудники
- Потеря исходных кодов
- Бардак на техподдержке
- Неравномерная загрузка разработчиков

Вторые проблемы

- Появился отдел маркетинга с кучей идей
- Продуктов стало очень много и все они разные
- Разработчики тормозят с выдачей результатов по непонятной причине
- Акционерам надоело заниматься всем, включая техподдержку 😊

Общий репозиторий исходников

- На тот момент популярными системами были CVS, VSS, SVN
- Все системы можно использовать с Visual Studio.
- Для подписчиков MSDN на тот момент Visual Studio была бесплатной
- Среда разработки: Visual Studio 2008
- SVN как общий репозиторий исходников
- Не только лишь все имеют доступ к чужим исходникам 😊

Проблемы при внедрении SVN

- Разработчики не понимают, зачем это нужно
- Разработчики думают, что над ними навис тотальный контроль
- Крайне редко делается Commit исходников
- Разработчикам лень прочитать документацию: неправильное использование

SVN: решение проблем

- На собеседовании с новыми разработчиками убеждаемся в знании SVN и подобных систем
- Обязательно используем SVN сами (акционеры все еще иногда пишут код)
- Не принимаем версию для релиза, пока не будет сделан Commit
- Смотрим и контролируем все движения в SVN
- В случае проблем объясняем, даем ссылки на документацию
- *Если строить фирму с нуля, этих проблем можно избежать сразу...*

Почему не git или Mercurial?

- Главный аргумент: оно работает без проблем и решает все задачи
- 90% разработчиков устраивает SVN
- Разработчикам не нужен локальный репозиторий
- Commit можно делать только локально. Commit на сервер – отдельное действие

От отдельных работников к команде

- Команда – это прежде всего люди
- Должна быть понятна цель
- Члены команды должны видеть результат работы
- Живое общение обязательно

Разработчик не только пишет код

- Прямое общение с клиентами в случае сложных случаев на техподдержке
- Участие в выставках и конференциях, общение с клиентами
- Главная проблема: английский язык

Немного о техподдержке

- Внедрили Kayako
- ВСЕ общение с клиентами только через тикеты
- Входящего E-Mail адреса у техподдержки нет
- Если как-то просочилось письмо – заводим тикет сами
- Sales и marketing тоже общаются через Kayako



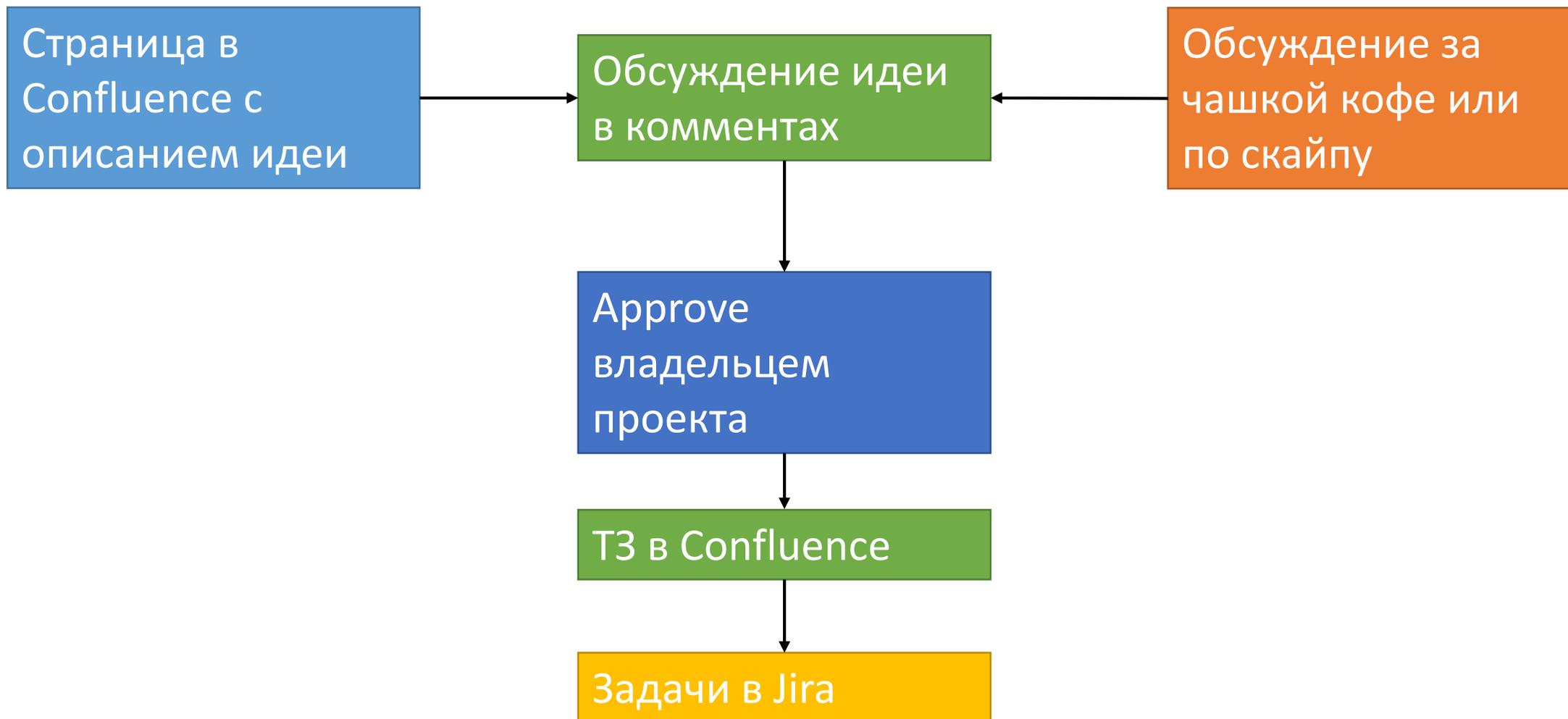
Хранилище знаний и управление проектами

- Confluence – продвинутая wiki
- Jira – система постановки задач
- Если ими пользоваться, бардак с кучей писем уходит
- Дешево для маленькой команды, дорого для большой

 Confluence

 Jira Software

От идеи к реализации



Jira: особенности использования

- Все члены команды имеют равные права
- Каждый может: создать задачу, передать задачу, комментировать и даже закрыть задачу
- Самый простой workflow:

CURRENT: Simple Issue Tracking Workflow

From	Transition	To
TO DO	Start Progress No Screen	→ IN PROGRESS
	Done  Resolve Issue Screen	→ DONE
IN PROGRESS	Stop Progress No Screen	→ TO DO
	Done  Resolve Issue Screen	→ DONE
DONE	Reopen No Screen	→ TO DO
	Reopen and start progress No Screen	→ IN PROGRESS

Проблемы

- Междусобойчики в офисе за чашкой чая, о которых никто не знает
- Длинные обсуждения по емайлу
- Попытки поставить задачу не через Jira
- Попытки начать что-то делать без ТЗ и задачи
- Решение: просто дайте команде время. Они сами поймут, что по-новому работать удобнее.