

# Инструменты командной работы

Конфигурационное управление

Екатерина Вершинина,  
Сергей Нечаев  
ООО «Новотелеком»

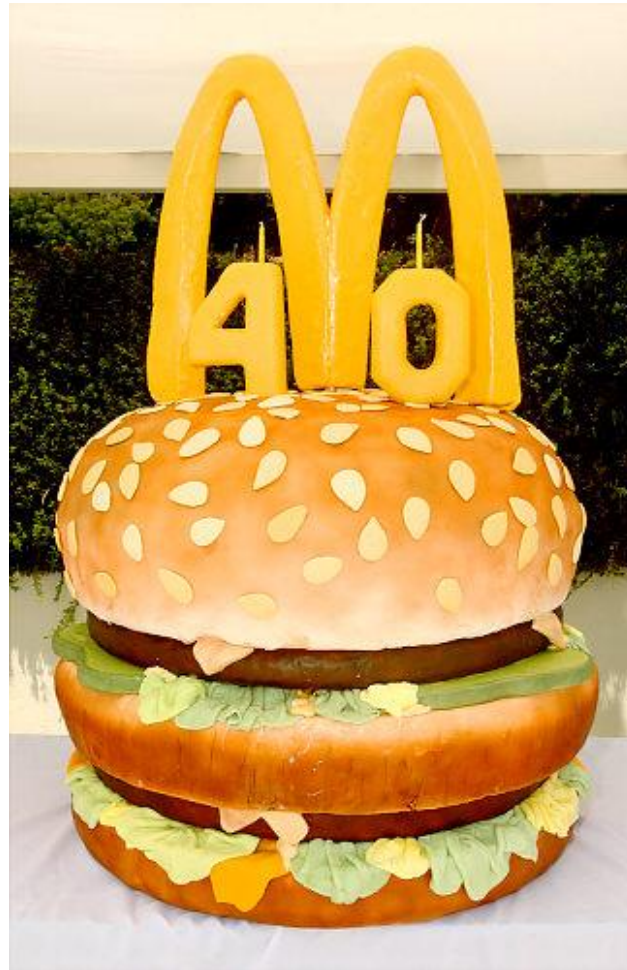
# План

- Кто все эти люди?
- Регламенты – наши друзья
- База знаний
- Issue trackers
- Системы контроля версий

# Кто все эти люди?!

- Хаос:
  - Не фиксированы требования (а что вообще делаем-то)
  - Неизвестно, кто что должен делать и кто чем занят
  - Где результат? Будет ли он? Когда?
  - Как пользоваться результатом?

# Регламенты – наши друзья



# Когда регламенты исполняются (добровольно)

- Регламент – фиксация договоренности.
- Регламенты исполняются, если:
  - Им проще следовать, чем не следовать
  - Их смысл понятен

# Что нужно регламентировать

- Роли и обязанности
  - Роль, что делает, за что отвечает.
- Правила работы с требованиями
  - Где требования лежат.
  - Что в них должно быть.
- Правила работы с кодом
  - Соглашения о кодировании
  - Архитектурный контроль
  - Просмотры кода для критичных участков

# Что нужно регламентировать - 2

- Правила работы с репозиторием
  - Структура
  - Коммиты
  - svn hooks или подобное
- Правила работы с issue tracker
  - Механизм определения приоритетов
  - Проекты, типы запросов и компоненты
- Правила работы с базой знаний
  - Что нужно документировать
  - Структура базы знаний
- Комплект проекта
  - Артефакты
  - Документация
  - План внедрения и отката (инструкции)

# Примеры регламентов

- Законы
- Войсковые уставы
- Правила дорожного движения
- Инструкции по технике безопасности



# Что не нужно регламентировать

- Помогай вам здравый смысл.



# Задачи тактического управления

- Взаимодействие с заказчиком и смежниками
- Постановка задач исполнителям
- Обеспечение и контроль исполнения
- Сохранение полученных результатов

# Взаимодействие с заказчиком

- На входе:
  - Неструктурированная информация
- На выходе:
  - Что нужно
  - Когда нужно
  - Критерии успеха
- А что дальше?

# База знаний

- Требования
- Документация
- Регламенты
- Накопленный опыт

# База знаний: варианты

- В голове
- На липких таких фигнях, которые приклеиваются
- В тетрадке или блокноте
- На доске
- В файле локально
- В файле на сервере
- В репозиториях
- В wiki системах

# Требования к базе знаний

- Простота внесения изменений
- Быстрое распространение изменений (как только появилось что-то новое, заинтересованные лица об этом узнали.)
- Долговечность.

# Wiki

- Mediawiki
- Wackowicki
- Atlassian confluence
- Trac
- ...

# Организация базы знаний

- Главная страница:
  - Что здесь есть
  - Как искать
  - Как добавлять
  - Ссылка на инструкцию по работе с wiki
  - Разделы:



# База знаний: разделы

- Описание подразделений
  - Сотрудники: роли, обязанности, ответственность
  - Задачи в работе и архив выполненных.
- Описание проектов
  - Архитектура, сопроводительная документация, требования и прочая
- Общие регламенты
  - О них было сказано
- Общие полезные советы
  - Грабли
  - Алгоритмы
  - Хитрые команды.

# Issue trackers

- Управление задачами и сотрудниками
  - Постановка задач
  - Контроль исполнения
  - Контроль занятости
  - Статистика (кто что когда-либо делал)
- Примеры
  - JIRA
  - Bugzilla
  - Trac
  - Mantis
  - ....

# Тикет – единица работ

- Автор                      кто создал тикет
- Название                      краткое описание задачи
- Описание                      детальное описание задачи
- Приоритет                      относительно других
- Срочность                      когда нужно сделать
- Исполнитель                      кто делает и отвечает
- Статус                      открыт/в работе/решен/закрыт
- Связи                      блокирует/зависит от/связан/дубликат

# Пример: тикет на уборку лужи

- Автор: Василий Пупкин
- Название: «В коридоре второго этажа разлита вода»
- Описание: «Между 201 и 203 кабинетами лужа. В нее легко случайно наступить. Просьба вытереть»
- Приоритет: **нормальный**.
- Срочность: сегодня
- Исполнитель: тетя Дуся.
- Статус: Открыт.

# тикет на уборку лужи

- Автор: Василий Пупкин
- Название: «В коридоре второго этаже разлита вода»
- Описание: «Между 201 и 203 кабинетами лужа. В нее легко случайно наступить. Просьба вытереть»
- Приоритет: **нормальный**.
- Срочность: прямо сейчас.
- Исполнитель: тетя Василиса.
- Статус **решается**.

# тикет на уборку лужи

- Тетя Василиса: «В туалете второго этажа прорвало трубу. Не имеет смысла вытирать – вода все равно набежит. Отложен до решения блокирующего тикета»
- Статус: **отложен**
- Создается новый тикет на сантехника, который блокирует тикет на уборку лужи.

# Тикет на сантехника

- Автор: Тетя Василиса
- Название: «В туалете между 201 и 203 кабинетом прорвало трубу с холодной водой»
- Описание: «В туалете между 201 и 203 кабинетом прорвало трубу с холодной водой. Пока не сильно течет, но труба может лопнуть совсем и тогда будет потоп. Надо починить»
- Приоритет: **Блокирующий**.
- Срочность: прямо сейчас
- Исполнитель: дядя Харитон.
- Статус: Открыт.

# Тикет на сантехника

- Дядя Харитон: «участок трубы холодного водоснабжения в туалете 2го этажа заменен, вода больше не течет»
- Статус тикета: «**решен**»



# Тикет на сантехника

- Тетя Василиса: «Спасибо, Харитон, как всегда четко и оперативно 😊»
- Статус тикета: **«закрыт»**

# тикет на уборку лужи

- Тетя Василиса: «Водопровод починили, пошла вытирать лужу»
- Статус: **решается**

# тикет на уборку лужи

- Тетя Василиса: «готово»
- Статус: **решен**

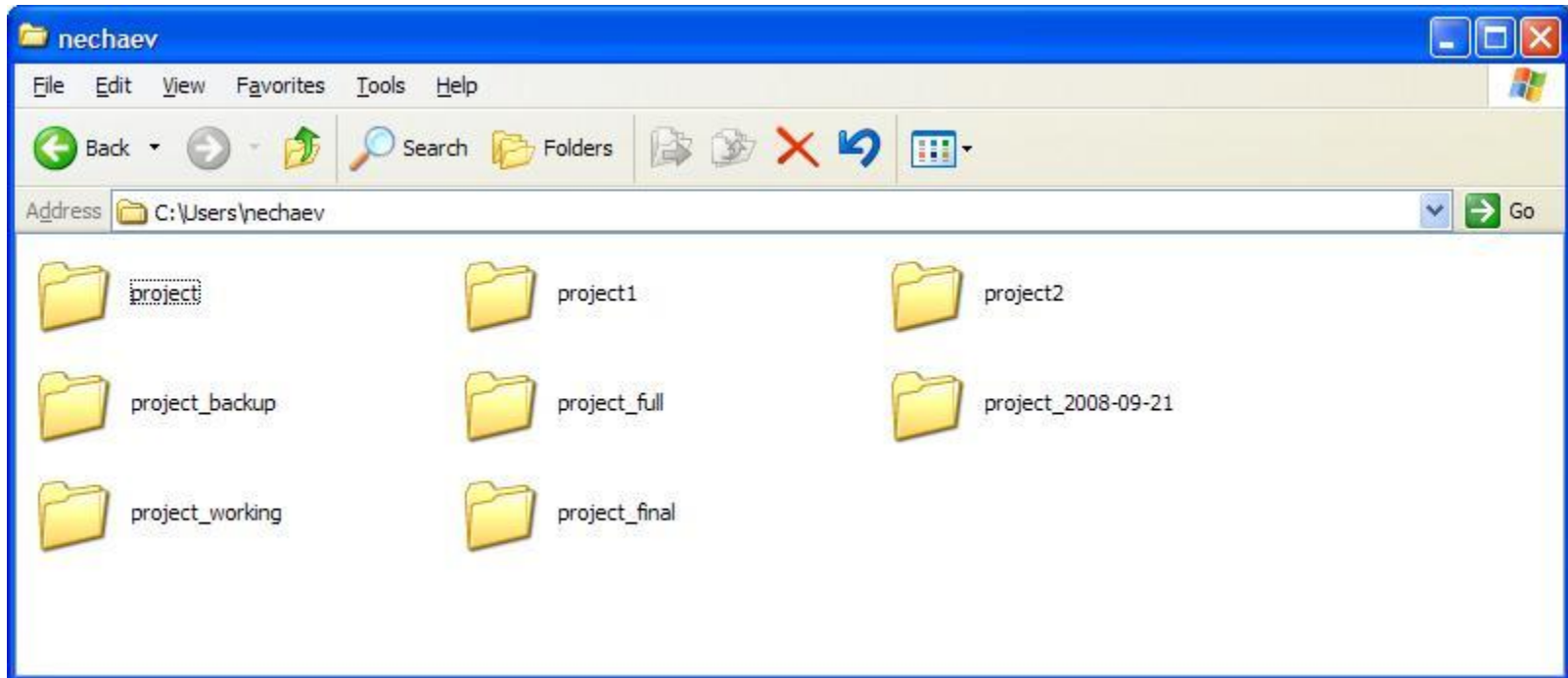
# тикет на уборку лужи

- Василий Пупкин: «Всем спасибо. Все молодцы!»
- Статус: **закрыт.**

# Советы

- Отдельные проекты:
  - Для общения с заказчиками:
    - Сроки по обязательствам
    - Контрольные точки
    - Запрос информации
  - Для общения с исполнителями
    - Сроки по внутренним планам
    - Детальные поручения с техническими подробностями
    - Контроль занятости группы

# Системы контроля версий



# Системы контроля версий

- Обеспечение совместной работы
  - Хранение файлов и истории их изменений
  - Кто что сделал
  - Возможность быстрого отката к работающей конфигурации
  - Объединение изменений
  - И прочая

# Примеры

- Централизованные (Клиент-сервер):
  - CVS
  - Perforce
  - Subversion (SVN)
  - TFS
  - ...
- Децентрализованные:
  - git



# Клиент-сервер

- Файло лежит на сервере
- Для внесения изменений:
  - Скачиваем файл(ы)
  - Локально редактируем
  - Пробуем закачать на сервер
    - Смотрим, нет ли конфликтов
    - Разрешаем конфликты
    - Закачиваем изменения на сервер

# Структура репозитория

- trunk
  - Основной каталог проекта
- Tags
  - Метки. Каждая метка это «слепок» ствола или релизной ветки на какой-то момент времени.
- Branches
  - Ветки. Могут быть:
    - «Релизными»: сопровождение текущей версии
    - «Функциональными»: добавление нового функционала
    - «прототипными»: прототипирование

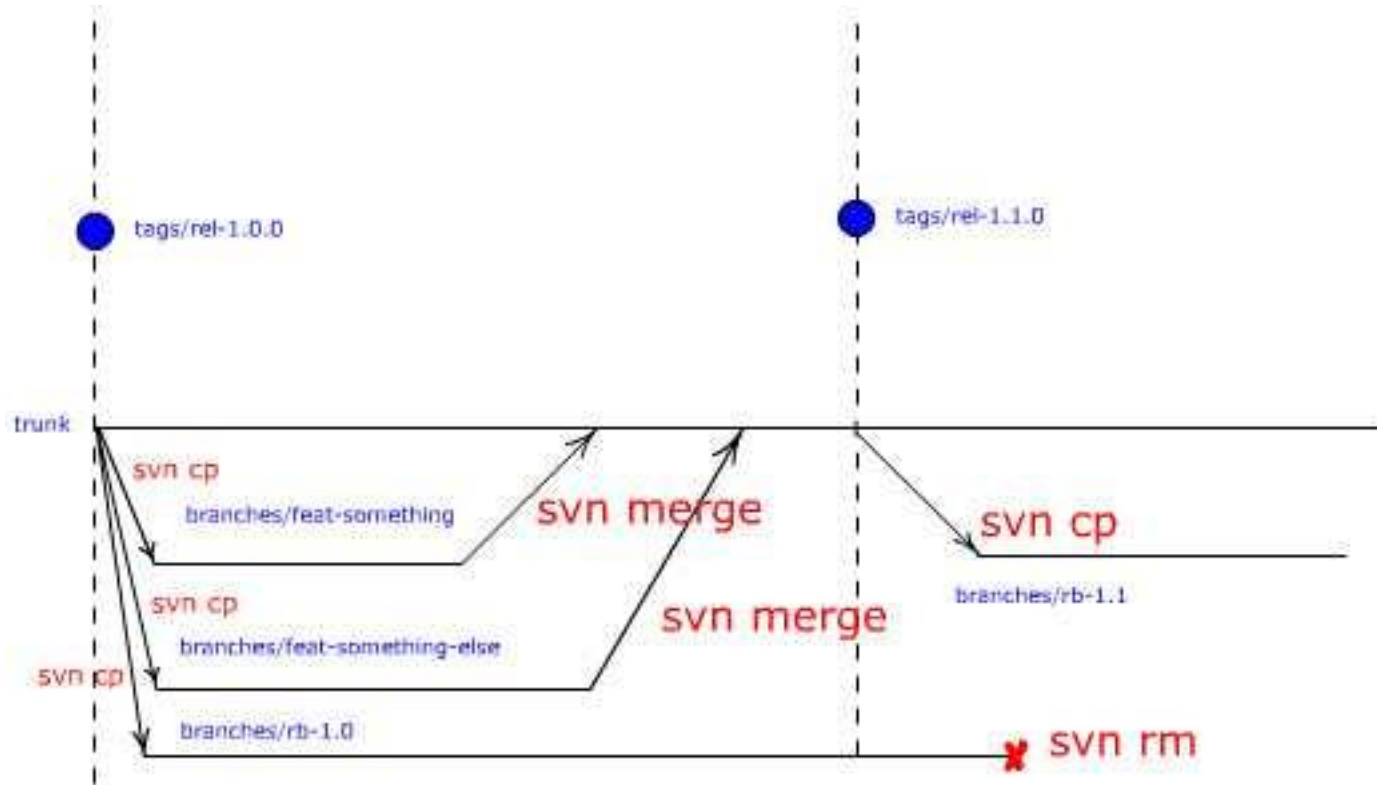
# Словарь

- **branch**
  - Ветвь — направление разработки, независимое от других. Документы в разных ветвях имеют одинаковую историю до точки ветвления и разные — после неё.
- **check-in, commit, submit**
  - Распространение изменений, сделанных в рабочей копии, на хранилище документов.
- **check-out, clone**
  - Извлечение документа из хранилища и создание рабочей копии.
- **conflict**
  - Конфликт — ситуация, когда несколько пользователей сделали изменения одного и того же участка документа..
- **merge, integration**
  - Слияние — объединение независимых изменений в единую версию документа. Осуществляется, когда два человека изменили один и тот же файл или при переносе изменений из одной ветки в другую.
- **repository**
  - Хранилище документов — место, где система управления версиями хранит все документы вместе с историей их изменения и другой служебной информацией.
- **revision**
  - Версия документа. Системы управления версиями различают версии по номерам, которые назначаются автоматически.
- **tag, label**
  - Метка, которую можно присвоить определённой версии документа. Ревизии включенных в метку документов могут принадлежать разным моментам времени.
- **trunk, mainline**
  - Ствол — основная ветвь разработки проекта.
- **update, sync**
  - Синхронизация рабочей копии до некоторого заданного состояния хранилища.
- **workingcopy**
  - Рабочая (локальная) копия документов.

# Пример нумерации версий

- N.M.K
  - N – major version number. Увеличивается при коренной переработке приложения, возможно, с потерей обратной совместимости
  - M – minor version number. Увеличивается при добавлении новой функциональности с сохранением обратной совместимости
  - K – bugfix version number. Функционал не меняется, только исправляются ошибки
- При увеличении номера версии, младшие номера обнуляются. 1.3.10 может стать:
  - 2.0.0
  - 1.4.0
  - 1.3.11

# Пример: 1.0.0. □ 1.1.0



# ОСНОВНЫЕ КОМАНДЫ

- Развертывание рабочей копии
  - **svn co <URL>**
- Работа в рабочей копии
  - **svn add <path>** - добавление
  - **svn rm <path>** - удаление
  - **svn mv <oldpath> <newpath>** - перемещение
  - **svn up <path>** получение изменений с сервера
  - **svn revert <path>** отмена локальных изменений
- Заливка изменений на сервер
  - **svn ci**
- Слияние и разрешение конфликтов
  - **svn merge** слияние
  - **svn resolve** отметка конфликта, как разрешенного
- Просмотр изменений
  - **svn log** просмотр истории изменений
  - **svn blame** построчный просмотр файла с указанием автора последних изменений каждой строки

# Правила хорошего тона (регламенты)

- Почтовые уведомления о коммитах группе разработки
- В метки не заливаться
- Информативные сообщения о коммитах. (желательно номер тикета указывать, по которому произведена заливка)
- В trunk – только собирающийся код

# Итоги

- Джентльменский набор IT-борца с хаосом:
  - Регламенты
  - База знаний
  - Issue Tracker
  - Система контроля версий



# Домашнее задание

- По всему перечисленному:
  - Знать и уметь объяснить зимним школьникам, что такое и зачем нужно.
  - Научиться пользоваться самим и уметь научить зимних школьников.
- На ЗШ использование перечисленных инструментов является обязательным (это регламент)

Вопросы? Замечания?

Внемлю.