**Лекция 1. Системы контроля версий**

**Система контроля версий (СКВ)** — ПО, призванное автоматизировать работу с историей файлов, обеспечить мониторинг изменений, синхронизацию данных, а также организовать защищённое хранилище проектов.

*Основная задача СКВ* — упростить работу с изменяющейся информацией.

**Основные возможности СКВ**

1. Позволяет проектной команде одновременно работать над одним и тем же проектом.

2. Минимизирует конфликты между изменениями, вносимыми разными разработчиками.

3. Автоматическое создание архивов каждой версии.

**Основные понятия и определения**

**• Репозиторий** — хранилище документов. Место, где СКВ хранит все файлы проекта вместе с дополнительной служебной информацией.

**• Рабочая копия (РК)** — локальная копия документов из хранилища.

**• Клонирование (сhесk-out/clone)** — извлечение документа из хранилища и создание рабочей копии.

**• Фиксация (спеck-in/commit)** — публикация локальных изменений в репозиторий.

**• Ветвь (branch)** — направление разработки, независимое от других.

**• Слияние (merge)** — объединения независимых изменений в единую версию документа.

**Виды СКВ**

**• Локальные СКВ** — СКВ, хранящиеся только на локальной машине разработчика.

**• Централизованные СКВ (ЦСКВ)** — системы, в которых работа производится с центральным хранилищем. Т.е. все действия, так или иначе, зависят от него.

**• Децентрализованные (распределённые) СКВ** — СКВ, основной парадигмой которой является локализация данных на машине каждого разработчика.

В РСКВ всякий раз, когда клиент забирает свежую версию файлов, он получает репозиторий целиком.

**Примеры СКВ**

1. CVS (ЦСКВ)

2. SVN (ЦСКВ)

3. GIT (ДСКВ)

4. Mercurial (ДСКВ)

5. Bazaar (ЦСКВ и ДСКВ)

**Лекция 2. Subversion. Основные сведения**

**Subversion (SVN)** — это свободно распространяемая ЦСКВ с открытым исходным кодом.

*Цель SVN* — заменить собой CVS и устранить имеющиеся у неё недостатки.

**Принцип работы SVN**

В процессе работы над проектом SVN запоминает каждое вносимое изменение. Любое изменение любого файла, также как и изменение структурных каталогов (удаление, добавление, копирование), запоминается и фиксируется системой.

При чтении данных из хранилища по умолчанию клиент видит последнюю версию файла, но также имеет возможность просматривать и предыдущие состояния файловой системы.

**Рабочая копия в SVN**

Рабочая копия в SVN представляет собой обычное дерево каталогов на локальном компьютере. Данные из PK недоступны для других участников проекта до момента их фиксации в хранилище.

**Служебный каталы PK в SVN**

В каждой РК содержится каталог с именем *.svn*, которой помогает SVN отслеживать текущее состояние РК. СК позволяет определить какие файлы РК содержит неопубликованные изменения, а какие файлы устарели по отношению к файлам других участников проекта.

Для каждого файла из РК SVN записывает информацию о 2 важнейших свойствах:

1. На какой правке (версии) основан ваш рабочий файл.

2. О временной метке, определяющей, когда РК последний раз обновлялась из хранилища.

Используя данную информацию, SVN может однозначно определить в каком из состояний находится каждый файл:

1. Не изменялся локально и не устарел.

2. Изменялся локально и не устарел.

3. Не изменялся и устарел.

4. Изменялся локально и устарел.

**Модели версионирования**

1) Блокирующая модель (блокирование — изменение — разблокирование)

2) Модель слияния (копирование — изменение — слияние).

Как правило, СКВ помогают в процессе слияния, но за его корректное выполнение всегда отвечает человек.

**Хранение изменений в SVN**

Дельта - кодирование — способ представления данных в виде разницы (дельты) между последовательными данными вместо самих данных.

**Жизненный цикл проекта с использованием SVN**

Просмотр справки по необходимой команде в SVN выполняется с помощью команды [svn help], где в качестве аргумента указывается имя требуемой команды.

**Создание репозитория в SVN**

**•** Для создания пустого репозитория SVN используется команда [svnadmin create] и дальше указывается путь до репозитория в текущей файловой системе.

**•** Для импортирования нового проекта в хранилище используется команда [svn import], при этом в качестве параметра указывается путь до импортируемого дерева файлов и каталог хранилища, в который необходимо выполнить импорт.

**• Создание РК.** Для создания РК уже существующего репозитория используется команда [svn checkout], в качестве параметра которой указывается путь до репозитория.

В зависимости от того, где находится SVN репозиторий, к нему можно получить доступ различными способами. При этом в URL репозитория указывается протокол доступа ([file:///](NULL)).

**Получение информации о текущей РК и репозитории**

Для получения сведений по текущей РК используется команда [svn info]. Она позволяет просматривать информацию о репозитории, такую как: его URL, номер текущей ревизии, дату последнего изменения и др.

Вызов команды необходимо осуществлять из директории, являющейся РК.

**Обновление РК**

Для обновления РК до последней версии хранилища используется команда [svn update], при этом будет произведена синхронизация содержимого РК с последней правкой хранилища. При этом для каждого элемента выводится специальный буквенный код, определяющий какое действие SVN произвёл с данным файлом.

**Внесение изменений в РК**

В SVN нельзя использовать обычные команды ОС для добавления, удаления и перемещения файлов. Для этого используются команды [svn copy/delete/move], т.к. только они записывают изменения в каталог *.svn*. В противном случае все внесённые изменения при обновлении будут потеряны.

**Основные команды**

**•** [svn add]

Служит для того, чтобы запланировать файл/директорию или символьную ссылку для добавления в хранилище при следующей фиксации (для добавления файла в отслеживаемые).

**•** [svn delete]

Предназначена для того, чтобы запланировать удаление из хранилища директории или символьной ссылки. Если удаляемый элемент является файлом, то он будет сразу же удалён из РК. Если удаляется директория, то она планируется для удаления и будет удалена из хранилища при фиксации изменений.

**•** [svn copy] [svn cope first.txt second.txt]

Создаёт новый элемент как копию первого. При этом скопированный элемент будет запланирован на добавление.

**•** [svn move]

Команда полностью аналогична последовательному использованию выполнению операций копирования и последующего удаления исходного элемента.