

УПП

Контроль версий

Лекция №4 (версия 1.0)

```
sent"/>  
fish.web.present
```

```
<!-- do not forg
```

```
oot}" else="{gfv
```

```
app.context-root]
```

```
resent">
```

```
b]"/>
```

Системы контроля версиями

Система управления/контроля версиями (от англ. Version Control System или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости, возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение и многое другое.

Системы контроля версиями

Такие системы наиболее широко применяются при разработке программного обеспечения, для хранения исходных кодов разрабатываемой программы. Однако, они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов, в частности, они всё чаще применяются в САПР, обычно, в составе систем управления данными об изделии (PDM). Управление версиями используется в инструментах конфигурационного управления (Software Configuration Management Tools).

Распространённые системы управления версиями

- **Subversion**
- Darcs
- **Microsoft Visual SourceSafe**
- Bazaar
- Rational ClearCase
- Perforce
- BitKeeper
- Mercurial
- **Git**
- GNU Arch
- **CVS** — устаревшая. Потомок: Subversion
- RCS — устаревшая. Потомок: CVS

ОСНОВНЫЕ ПОНЯТИЯ

Репозиторий (repository) – центральное хранилище, которое содержит версии файлов. Очень часто репозиторий организуется средствами какой-нибудь СУБД.

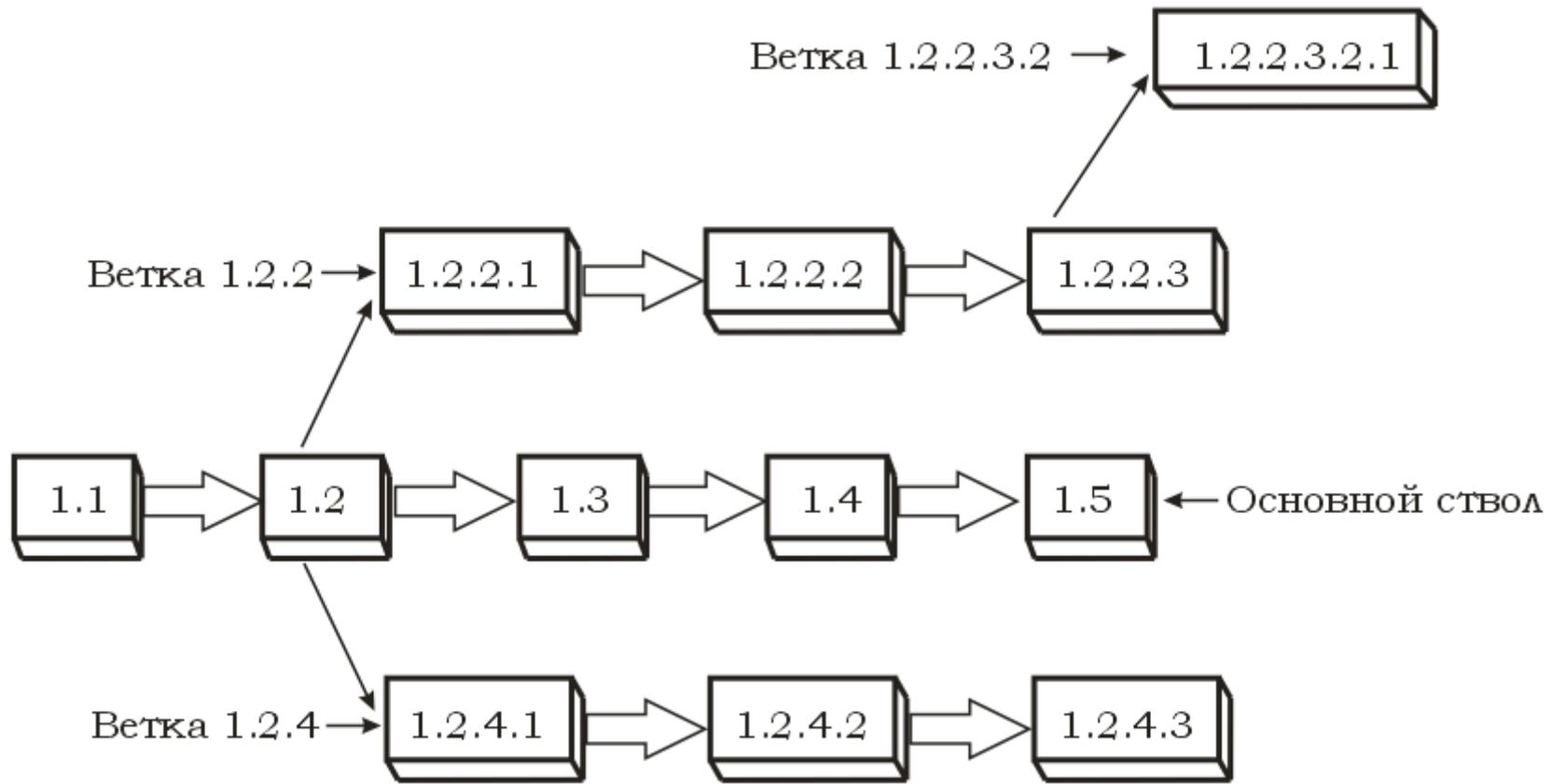
Версия файла (revision) – состояние файла в определенный момент времени. Репозиторий предоставляет возможность хранить неограниченное число версий одного и того же файла.

Актуальная версия файла – обычно это самая последняя версия файла, размещенного в репозитории.

Версионность

Что же такое редакция файла? В процессе работы над проектом файлы претерпевают изменения, добавляются новые и исчезают ненужные. Редакцией называется зафиксированное в репозитории (центральном хранилище файлов) состояние файла. Изменения файлов в рабочем каталоге не создают новых редакций, сколько бы дней или даже месяцев вы над ними не работали. Редакция появляется тогда, когда вы отправляете изменения в репозиторий. Самая первая редакция появляется при добавлении файла в репозиторий и получает номер 1.1.





Кроме цифровых номеров редакций, которые для простоты можно рассматривать как внутренний механизм CVS, можно использовать и символические имена. Мы можем присвоить символическое имя отдельной редакции отдельного файла, группе файлов или всему проекту сразу. Для выполнения этой функции используются «метки» (tags).

ОСНОВНЫЕ ПОНЯТИЯ

Рабочая версия файла (working copy) – версия файла, с которой в текущий момент ведется работа, и которая не загружена в репозиторий.

Загрузка (Upload) – размещение файла в репозитории. В процессе загрузки в репозиторий помещается рабочая версия файла.

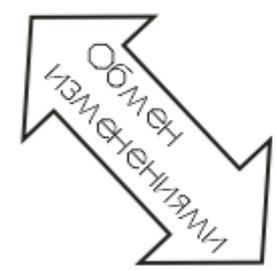
Выгрузка (Checkout) – получение файла из репозитория. В процессе выгрузки осуществляется получение из репозитория необходимой версии файла.

ОСНОВНЫЕ ПОНЯТИЯ

Синхронизация (update, sync) – приведение в соответствие рабочих версий файлов с актуальными версиями в репозитории. В процессе синхронизации в репозиторий загружаются те файлы, рабочие копии которых являются более "свежими" (т.е. имеют более поздние версии), по сравнению с файлами в репозитории, и выгружаются те файлы, рабочие копии которых устарели по сравнению с копиями в репозитории.

```
2 <project def:
3   <target r
4     <pro:
5     <ava:
6     <ava:
7     <ava:
8     <tem:
9     <ech:
10  </target>
11
12  <targ
13    <
14    <
15    <
16    <
17    <
18    <
19    <
20    <
21    <
22    <
23    <
24    <
25    <
26    <
27    <
28    <
29    <
30    <
31    <
32    <
33  </tar
34  <target n
35    <temp
36    <copy
```

```
sent"/>
fish.web.present
<!-- do not forg
```



```
else="$gfv
ontext-root,
```

```
resent">
b]"/>
```

Дополнительные понятия

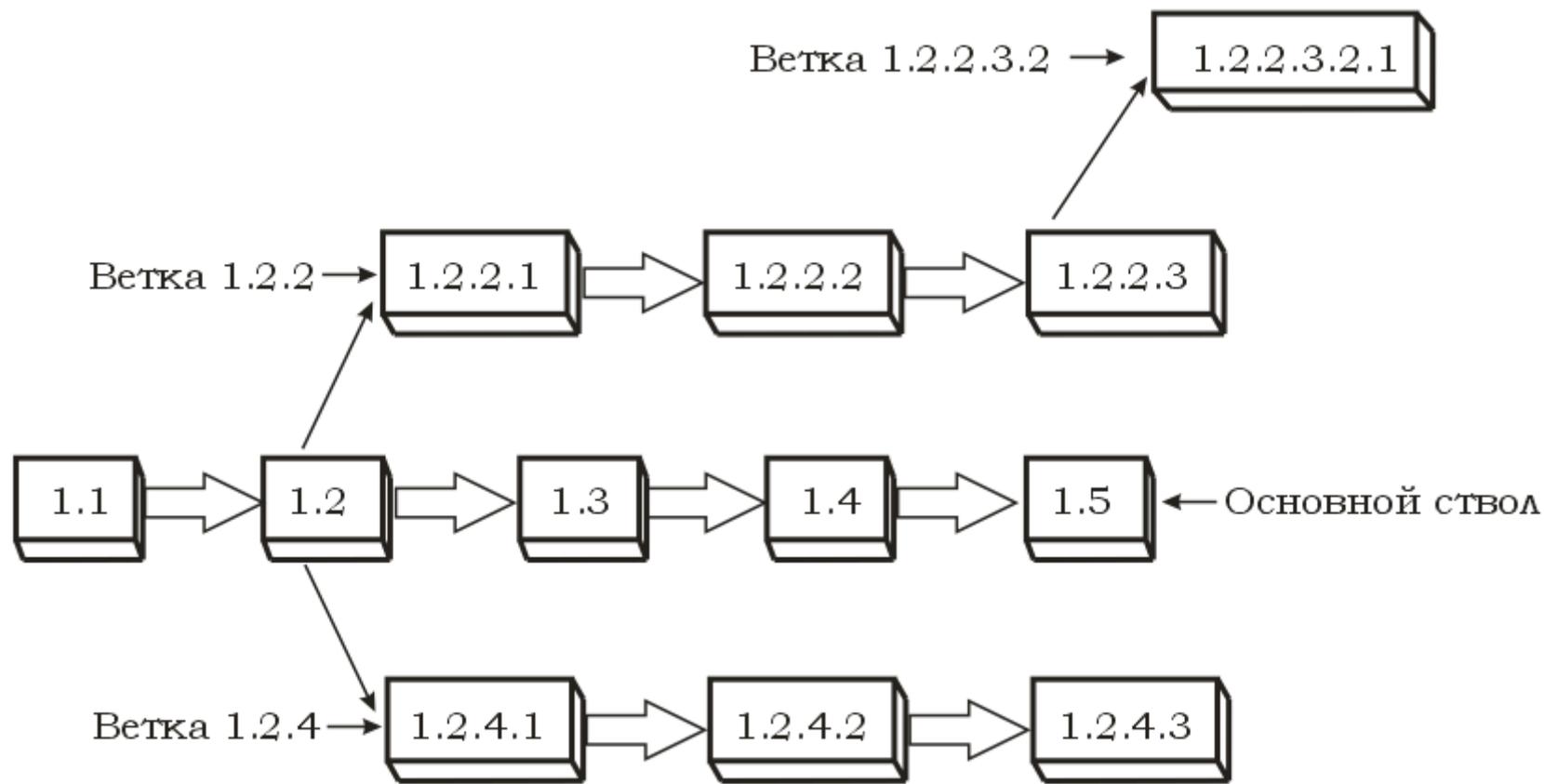
Ветвь (branch) — направление разработки, независимое от других. Ветвь представляет собой копию части (как правило, одного каталога) хранилища, в которую можно вносить свои изменения, не влияющие на другие ветви. Документы в разных ветвях имеют одинаковую историю до точки ветвления и разные — после неё.

Дополнительные понятия

Ствол (trunk, mainline) — основная ветвь разработки проекта. Политика работы со стволом может отличаться от проекта к проекту, но в целом она такова: большинство изменений вносится в ствол; если требуется серьёзное изменение, способное привести к нестабильности, создаётся ветвь, которое сливается со стволом, когда нововведение будет в достаточной мере испытано; перед выпуском очередной версии создаётся «релизная» ветвь, в которую вносятся только исправления.

```
2 <project def:
3   <target r
4     <pro
5     <ava
6     <ava
7     <ava
8     <tem
9     <ech
10  </tar
11
12  <tar
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33  </tar
34  <target n
35    <temp
36    <copy
```

```
sent"/>
fish.web.present
<!-- do not forg
```



```
else="$ {gf
context-root
```

```
resent">
b]"/>
```

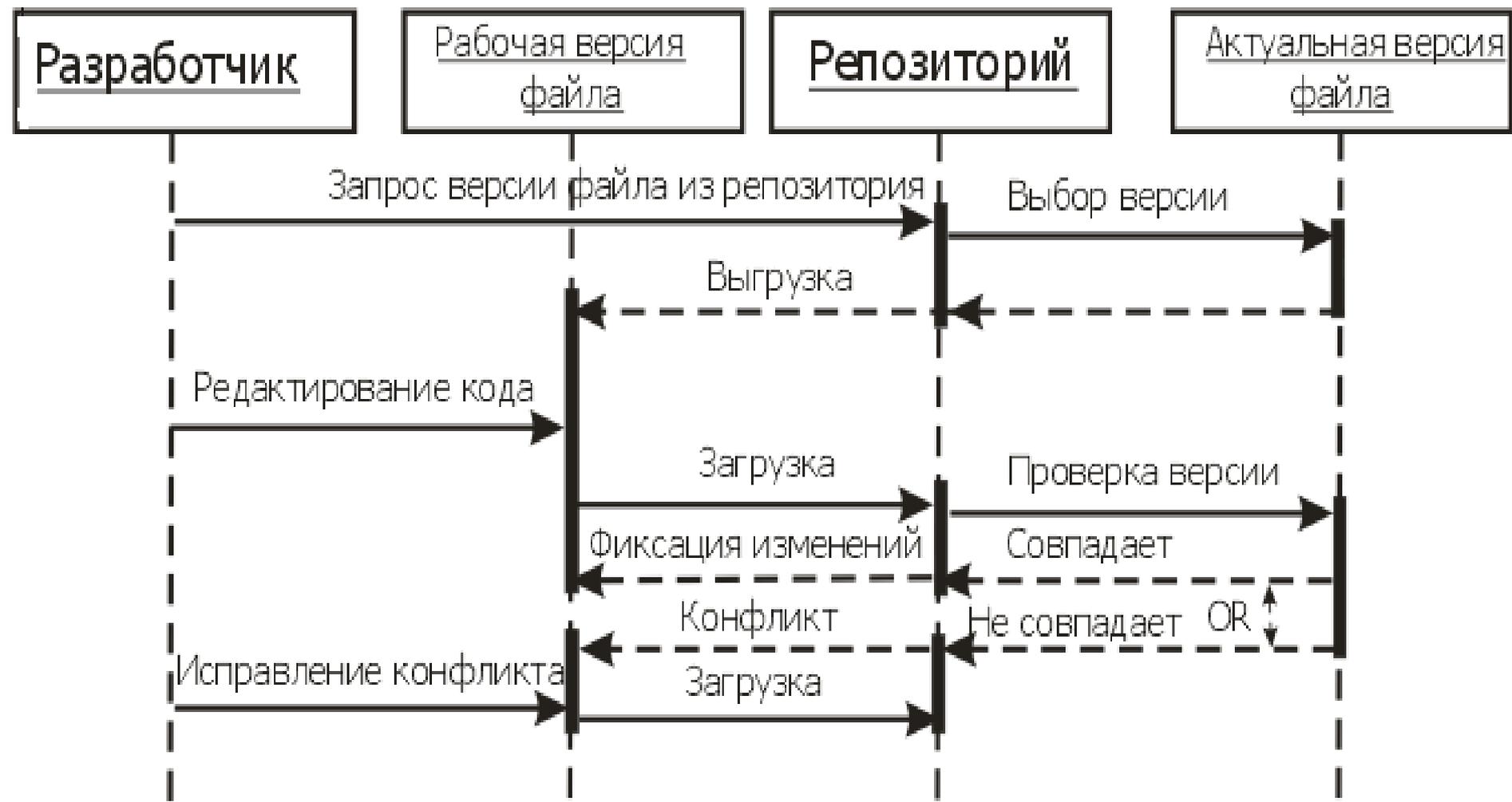
Дополнительные понятия

Метка (tag, label) - метка , которую можно присвоить определённой версии документа. Метка представляет собой символическое имя для группы документов, причем метка описывает не только набор имен файлов, но и ревизию каждого файла.

Конфликт — ситуация, когда несколько пользователей сделали изменения одного и того же участка документа. Конфликт обнаруживается, когда один пользователь опубликовал свои изменения, а второй пытается опубликовать и система сама не может корректно слить конфликтующие изменения. Второму пользователю нужно самому разрешить конфликт (resolve).

```
2 <project def:
3   <target :
4     <pro:
5     <ava:
6     <ava:
7     <ava:
8     <tem:
9     <ech:
10  </target:
11
```

```
sent"/>
fish.web.present
<!-- do not for
```



```
:"s(gf)
:-root)
```

Borland StarTeam

Borland StarTeam – очень мощный и функциональный кросс-платформенный продукт, разрабатываемый в прошлом фирмой StarBase, которую Borland приобрела в конце 2002 г. Заметное преимущество данного решения состоит в том, что версия 2005 выступает центральным элементом стратегии управления жизненным циклом приложений (Application Lifecycle Management, ALM) компании Borland и обладает расширенными возможностями интеграции со всеми ее ключевыми пакетами, используемыми при разработке программного обеспечения.

MS SourceSafe

Microsoft Visual SourceSafe (Visual SourceSafe, VSS) — программный продукт компании Майкрософт, файл-серверная система управления версиями, предназначенная для небольших команд разработчиков. VSS позволяет хранить в общем хранилище файлы, разделяемые несколькими пользователями, для каждого файла хранится история версий.

VSS входит в состав пакета Microsoft Visual Studio и интегрирован с продуктами этого пакета. Доступен только для платформы Windows. Версию для Unix поддерживает компания MainSoft.

В ноябре 2005 года вышла обновлённая версия продукта — Visual SourceSafe 2005, обещающая повышенную стабильность и производительность, улучшенный механизм слияния для XML-файлов и файлов в Юникоде, а также работу через HTTP.

Visual SourceSafe нацелен на индивидуальных разработчиков либо небольшие команды разработчиков. Там где VSS недостаточно, ему на замену предлагается новый продукт Майкрософт — Team Foundation Server, входящий в состав Visual Studio Team System.

Rational Clear Case

ClearCase поддерживает следующие возможности, разительно отличающие его в лучшую сторону от других средств контроля:

- Общий контроль версий не только файлов, но и директорий/поддиректорий;
- Бесконечное число ответвлений от определенной версии;
- Автоматическая компрессия файлов и их кеширование (CC позволяет хранить большое количество данных, при всем при этом база данных остается компактной и быстрой);
- Позволяет легко конвертировать базы данных других средств контроля, например: PVCS, SourceSafe, RCS, CVS и SCCS;
- Поддерживает параллельную разработку и мультикомандные подразделения, расположенные в географически удаленных друг от друга местах;
- Мультиплатформенность (способен объединить единой средой участников, работающих на разных операционных системах);
- Имеет интеграцию со средствами разработки;
- Имеет Web-интерфейс для удаленного контроля.

CVS

CVS (Concurrent Versions System, "Система Конкурирующих Версий"). Хранит историю изменений определённого набора файлов, как правило исходного кода программного обеспечения, и облегчает совместную работу группы людей (часто — программистов) над одним проектом. CVS популярна в мире открытого ПО. Система распространяется на условиях лицензии GNU GPL.

```
2 <project def:
3   <target y
4     <pro:
5     <ava:
6     <ava:
7     <ava:
8     <tem:
9     <ech:
10  </target:
```

```
sent"/>
fish.web.present
<!-- do not forg
```



```
:"S(gf)
:-root,
```

Subversion

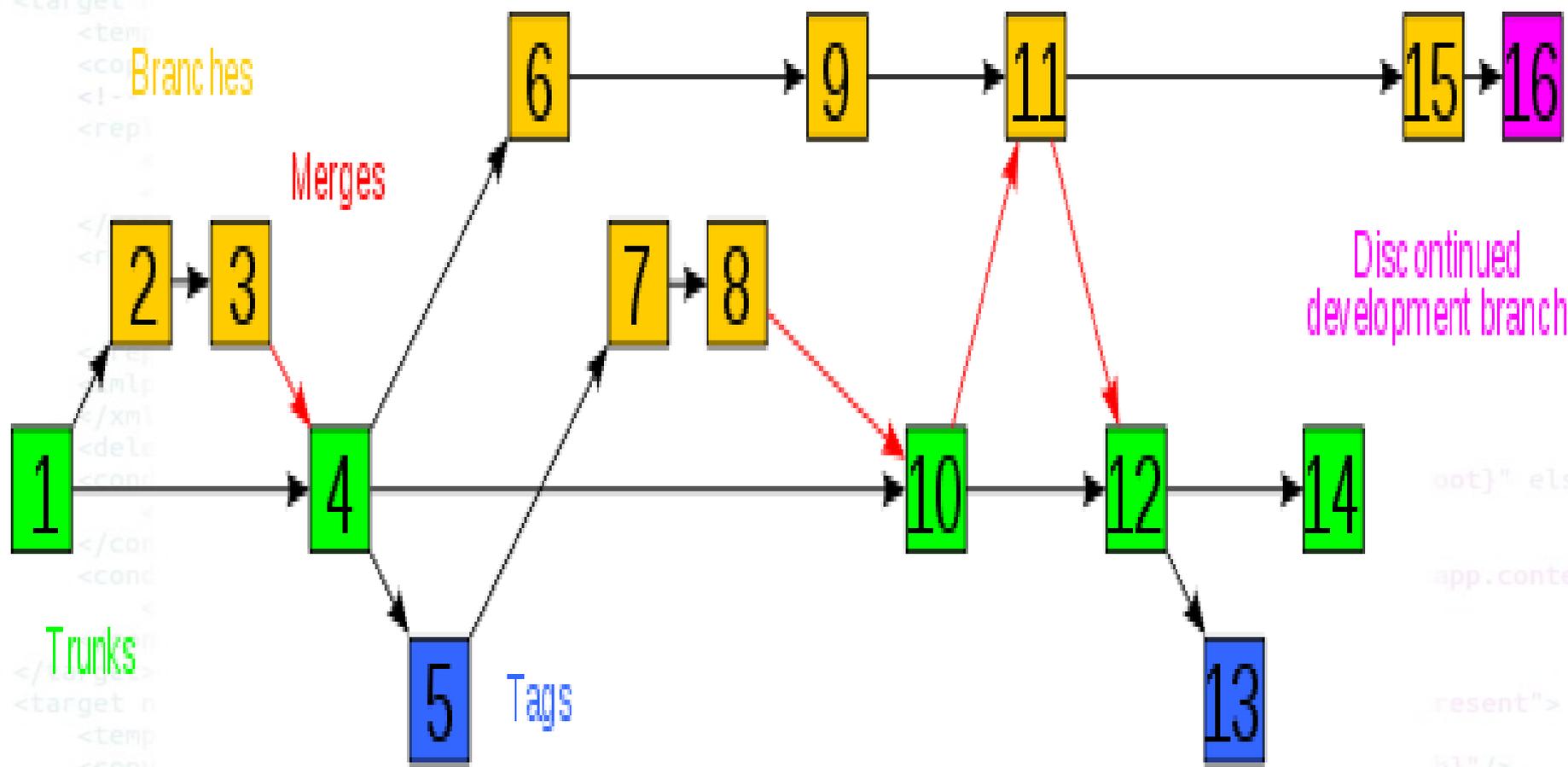
Subversion — централизованная система (в отличие от распределённых систем, таких, как Git или Mercurial), то есть данные хранятся в едином хранилище. Хранилище может располагаться на локальном диске или на сетевом сервере.

Работа в Subversion мало отличается от работы в других централизованных системах управления версиями. Для совместной работы над файлами в Subversion преимущественно используется модель Копирование-Изменение-Слияние. Кроме того, для файлов, не допускающих слияние (различные бинарные форматы файлов), можно использовать модель Блокирование-Изменение-Разблокирование.

Subversion (ВОЗМОЖНОСТИ)

- Реализовано большинство возможностей CVS.
- Отслеживается история файлов, директорий и метаданных файлов и директорий, в том числе при переименовании и копировании.
- Атомарная фиксация изменений.
- Возможность организации доступа к хранилищу Subversion через Apache по протоколу WebDAV/DeltaV.
- Возможность установки автономного сервера Subversion с доступом по собственному протоколу.
- «Дешёвые» операции создания ветвей и меток (требуется небольшое фиксированное количество временных и дисковых ресурсов).
- Многоуровневая архитектура библиотек, изначально рассчитанная на клиент-серверную модель.
- Клиент-серверный протокол разработан для пересылки по сети только разницы между объектами, когда это возможно.
- Затраты ресурсов пропорциональны размеру изменений, а не размеру данных, которые затронуты изменениями.
- Два возможных внутренних формата хранилища (англ. repository): база данных или набор обычных файлов.
- Версионированные символьные ссылки (только в рабочих копиях под UNIX-системами).
- Одинаково эффективная работа и с текстовыми, и с двоичными файлами.
- Вывод клиента командной строки одинаково удобен и для чтения, и для разбора программами.
- Интернационализированные сообщения программы (используются настройки локали).
- Библиотеки для языков PHP, Python, Perl, Java. Позволяют встроить функциональность клиента Subversion в программы, написанные на этих языках.
- Возможность зеркалирования хранилища.

Subversion (ветвление)

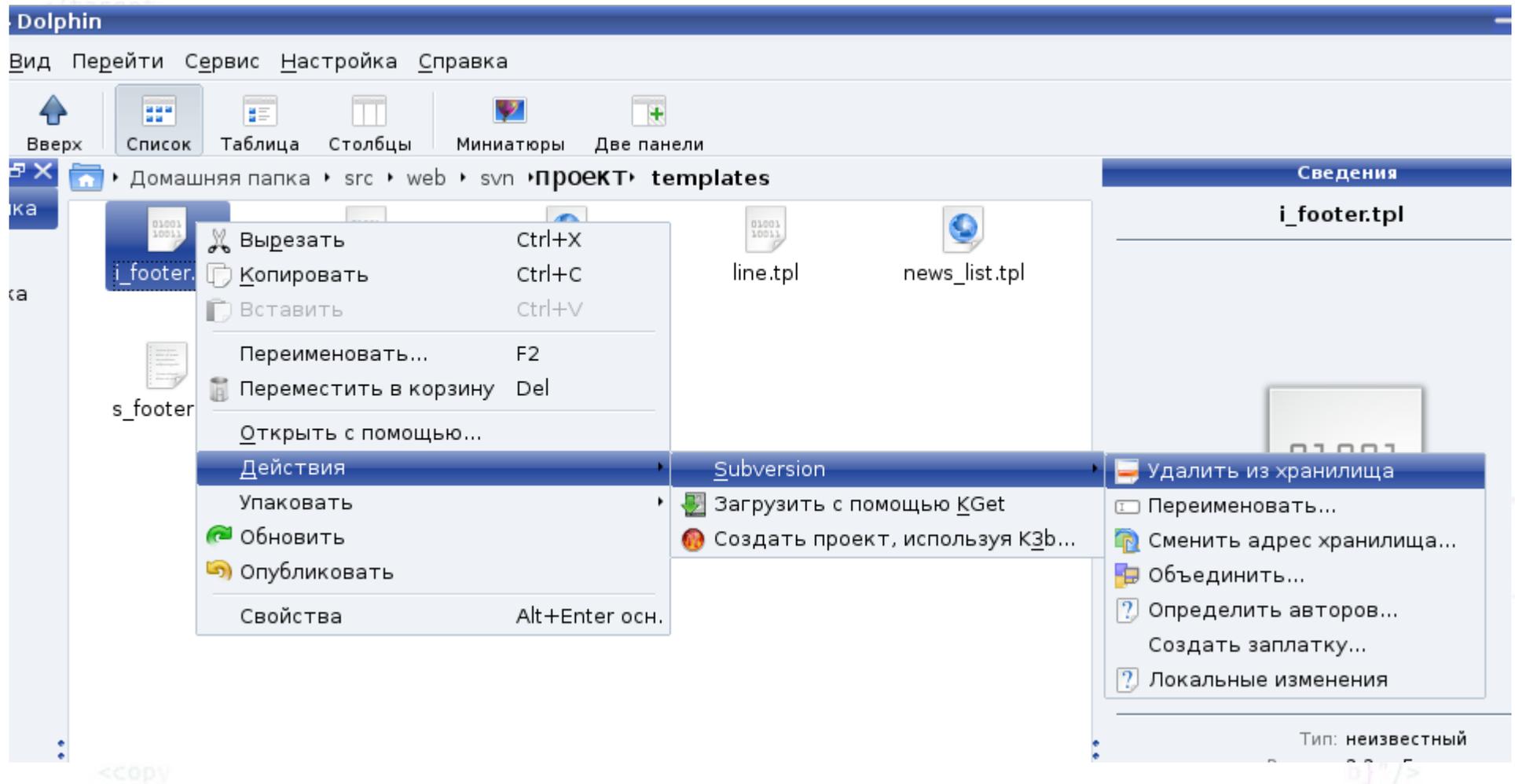


```
sent"/>  
fish.web.present  
<!-- do not forg
```

```
oot)" else="$gfv  
app.context-root
```

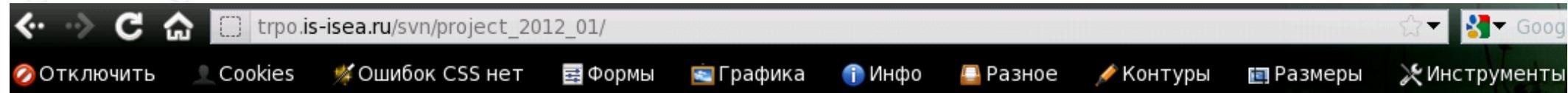
```
resent">  
<temp  
<copy
```

Работа с Subversion в Linux



2012

<http://elab.pro/svn/>



project_2012_01 - Revision 1: /

- [readme.txt](#)

Powered by [Subversion](#) version 1.6.12 (r955767).

2012

http://elab.pro/websvn/

```
2 <project def
3   <target r
4     <pro
5     <ava
6     <ava
7     <ava
8     <tem
9     <ech
10  </target>
11
12  <target r
13    <tem
14    <cop
15    <!--
16    <rep
17
```

```
sent"/>
fish.web.present
<!-- do not forg
```

trpo.is-isea.ru/websvn/listing.php?rename=project_2012_01&language=ru

Отключить Cookies Ошибок CSS нет Формы Графика Инфо Разное Контурь Размеры Инструменты Код Настройки

ХРАНИЛИЩА SUBVERSION PROJECT_2012_01 calm Русский - Russian

(root)/ - Редакция 1 Редакция HEAD Перейти

Последнее изменение | Открыть журнал | RSS

ПОСЛЕДНЕЕ ИЗМЕНЕНИЕ

Редакция 1 2012-04-02 23:57:10

Автор: user1201

Запись в журнале:

Путь	Последнее изменение	Журнал	RSS
<input type="checkbox"/> readme.txt	1 8ч 26м user1201	Журнал	RSS

Сравнить пути

2012

```
svn co http://elab.pro/svn/poject\_2012\_01/ --username  
      <username>
```

```
svn add <filename>
```

```
svn ci
```

```
sent"/>  
fish.web.present
```

```
<!-- do not forg
```

```
oot}" else="$ {gf
```

```
app.context-root
```

```
resent">
```

```
b]"/>
```

DVCS

Распределённые системы управления версиями

Такие системы используют распределённую модель вместо традиционной клиент-серверной. Они, в общем случае, не нуждаются в централизованном хранилище: вся история изменения документов хранится на каждом компьютере, в локальном хранилище, и при необходимости отдельные фрагменты истории локального хранилища синхронизируются с аналогичным хранилищем на другом компьютере. В некоторых таких системах локальное хранилище располагается непосредственно в каталогах рабочей копии.

Принципы работы с VCS

Для нового проекта

- Определить название репозитория и создать его
- Создать структуру каталогов
- Создать начальные файлы
- Добавить все файлы и каталоги в репозиторий

Для существующего проекта

- Выбрать место на диске для работы с проектом
- Подключиться к репозиторию и выгрузить проект на локальный компьютер

Во время работы

- Добавлять в репозиторий только работающие артефакты
- При фиксации изменений кратко описывать их суть
- Если VCS позволяет то группы файлов и каталогов заносить совместно