

# Технологии программирования

Часть 7

Направление:

Информатика и вычислительная техника  
Инфокоммуникационные технологии и системы связи

Ст. преподаватель  
кафедры ПИВТ  
Петрова О.Б.

2017 год

# Оценка качества процесса конструирования

- ISO 9001:2015 Системы менеджмента качества. Требования
- ISO/IEC 15504 Информационные технологии. Оценка процессов (Управление качеством разработки программного обеспечения)
- CMM (Capability Maturity Model) - эволюционная модель зрелости процессов создания программного обеспечения (ПО).
- Незрелой считается организация, применяющая процесс разработки ПО, зависящий только от конкретных исполнителей и определяющийся творческим подходом. Это приводит к трудностям в управлении проектами и их сроками
  - ◆ Начальный уровень (initial level).
  - ◆ Повторяемый уровень (repeatable level).
  - ◆ Определенный уровень (defined level).
  - ◆ Управляемый уровень (managed level).
  - ◆ Оптимизирующий уровень (optimizing level).

# Уровни зрелости организации

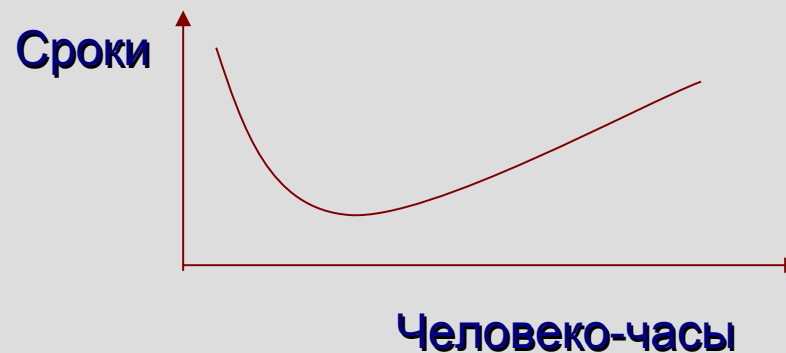
- ★ Начальный уровень. Организации, разрабатывающие ПО, но не имеющие осознанного процесса разработки, не производящие планирования и оценок, находятся на этом уровне.
- ★ Повторяемый уровень. В таких организациях ведется учет затрат ресурсов и отслеживается ход проектов, установлены правила управления проектами, основанные на имеющемся опыте.
- ★ Определенный уровень. В таких организациях имеется принятый, полностью документированный процесс разработки ПО, включающий как управленческие, так и технические процессы.
- ★ Управляемый уровень. В этих организациях, помимо установленного и описанного процесса, используются показатели качества продуктов и процессов, позволяющие достаточно точно предсказывать объем ресурсов (времени, денег, персонала), необходимый для разработки продукта с определенным качеством.
- ★ Оптимизирующий уровень. В таких организациях помимо процессов и методов их оценки, установлены процедуры поиска и оценки новых методов и техник разработки, обучения персонала работе с ними и их включения в общий процесс организации в случае повышения ими эффективности производства.

# Критерии качества программного продукта

- **ISO 9126 (ГОСТ Р ИСО / МЭК 9126-93) — «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению».**
- **Функциональность – соответствие требованиям пользователя**
- **Практичность – оценка возможности использования ПО**
- **Эффективность – соответствие между качеством ПО и используемыми ресурсами**
- **Сопровождаемость – оценка возможности внесения изменений в ПО**
- **Мобильность – возможность переноса ПО из одного окружения в другое**

# Планирование проекта

- Задача: выполнить требуемый объем работ в указанный срок с использованием ограниченного объема ресурсов
- Ресурсы:
  - ◆ Финансы
  - ◆ Люди (человеко-часы)
  - ◆ Материальные ресурсы
  - ◆ Нематериальные ресурсы
- Этапы планирования
  - Анализ риска
  - Планирование сроков: 40%-20%-40%
  - Планирование ресурсов



# Количественные оценки процесса разработки программного продукта

- Метрики – для планирования ресурсов
  - Размерно-ориентированные (LOC, Lines Of Code)
  - Функционально-ориентированные (FP – Function Points – функциональный указатель или Features Points – указатель свойств)
    - Характеристики полезности ПО:
      - ★ Количество внешних вводов
      - ★ Количество внешних выводов
      - ★ Количество внешних запросов
      - ★ Количество внутренних логических файлов
      - ★ Количество внешних интерфейсных файлов
      - ★ Количество алгоритмов

# LOC-метрики

- Производительность
  - ★ Длина кода/Затраты времени (тыс. LOC/чел.-мес)
- Качество
  - ★ Выявленные ошибки/Длина кода (ошибок/тыс. LOC)
- Удельная стоимость
  - ★ Стоимость/Длина кода (\$/тыс. LOC)
- Документированность
  - ★ Страниц документов/Длина кода (страниц/тыс. LOC)

# FP-метрики

- Производительность
  - ★ Функц. Ук./Затраты времени (FP/чел.-мес)
- Качество
  - ★ Выявленные ошибки/Функц. Ук. (ошибок/FP)
- Удельная стоимость
  - ★ Стоимость/Функц. Ук. (\$/FP)
- Документированность
  - ★ Страниц документов/Функц. Ук. (страниц/FP)
- Существует связь между LOC и FP метриками.
- Методики (модели) оценки ресурсов проекта – COSOMO (Гарри Боэм)

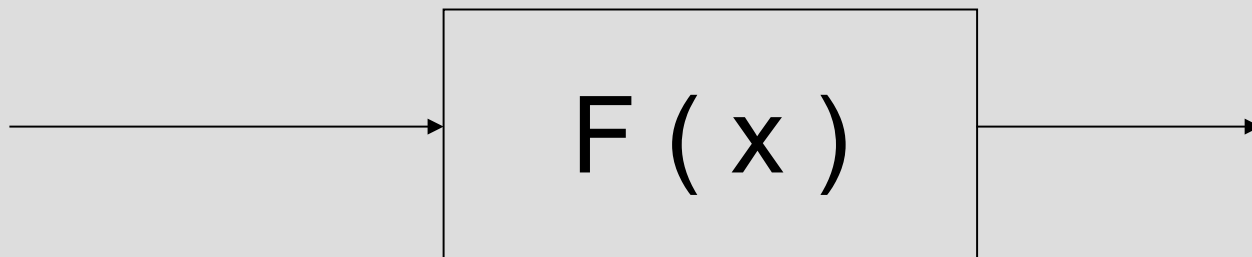


# Общие принципы тестирования

- Тестирование обнаруживает:
  - Ошибки в вычислениях
  - Логические ошибки
  - Несоответствие функций программы и ее назначения (неполнота функций)
  - Несоответствие функций программы требованиям заказчика
  - Частые отказы, сбои
- Тестирование не доказывает безошибочность программы!
- Принципы тестирования («черного ящика» и «белого ящика»)

# Тестирование «черного ящика»

- Используется для тестирования функций программы (интерфейса пользователя)
- Управляющая структура тестируемого кода неизвестна, а известно, какой должен быть результат



# Тестирование «белого ящика»

- Позволяет обнаружить ошибки в вычислениях и в логике
- Основано на анализе управляющей структуры программного кода
- Осуществляется проверка всех независимых маршрутов (путей)
- Недостаток: если путей много, тестирование трудоемко
- Пример: количество маршрутов для тестирования цикла с разветвлением

$k$

$m = \sum n^i$ , где  $k$  – количество проходов цикла,  $n$  – количество разветвлений в теле цикла,  $i=1..k$

$$k = 3 \quad n = 3 \quad m = 39$$

$$k = 20 \quad n = 5 \quad m = 10^{14}$$

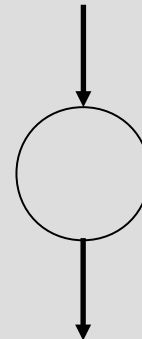
# Способы тестирования («белый ящик»)

- Тестирование базового пути
  - ◆ Позволяет оценить сложность программы
  - ◆ Позволяет определить количество тестовых вариантов
- Тестирование циклов
  - Тестирование простого цикла
  - Тестирование вложенного цикла

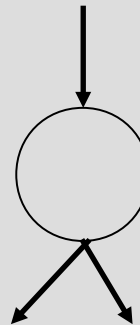
# Тестирование базового пути

- Для представления программы используется потоковый граф
- Узлы графа:

- ♦ Операторный

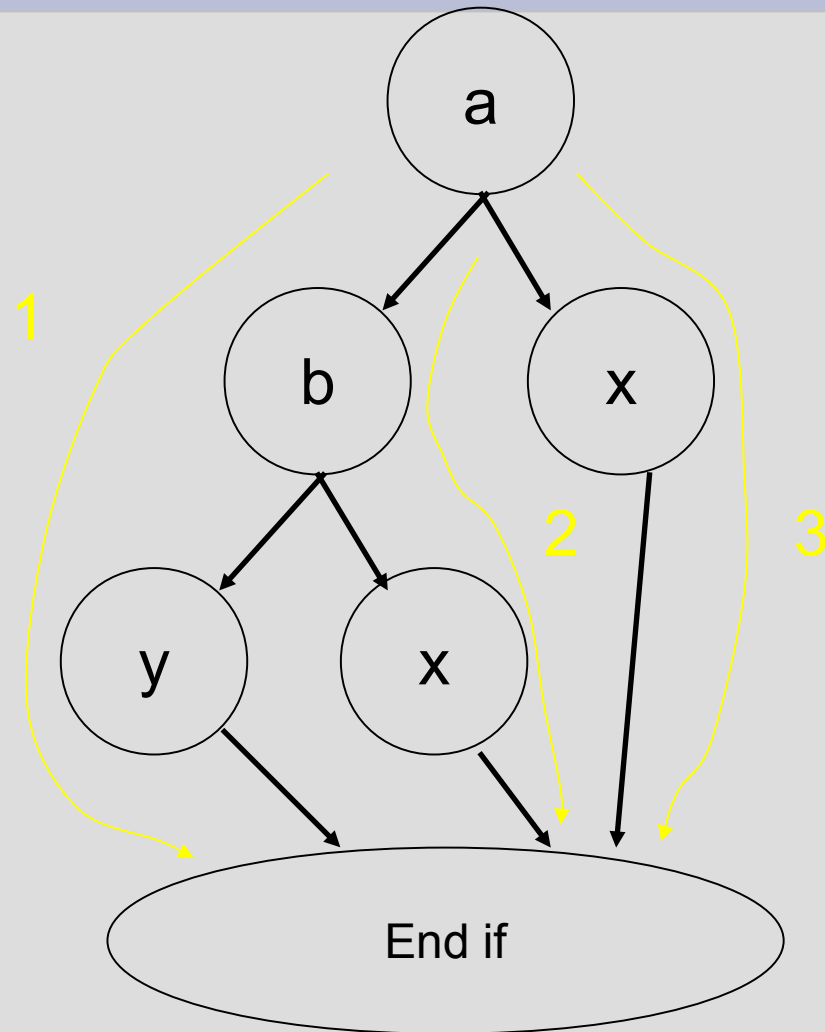


- ♦ Предикатный



# Пример

If a or b  
then x  
else y  
End if;

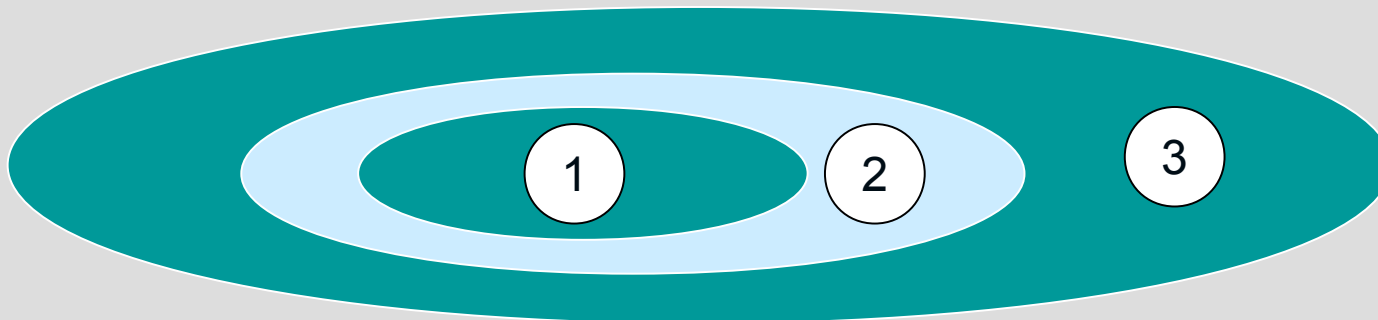


# Тестирование простого цикла

- Прогон всего цикла ( $n$ )
- Обход цикла
- Один проход цикла
- Два прохода цикла
- Прогон цикла, где  $m < n$
- $n - 1$ ,  $n + 1$  проходов цикла

# Тестирование вложенного цикла

- Тестирование внутреннего цикла, для внешних циклов устанавливаются минимальные значения параметров
- Тестирование следующего цикла, при этом сохраняются минимальные значения для внешних циклов и типовые (проверенные) значения для вложенных





# Методика тестирования программ

- Тестирование элементов (модулей, классов) – проверка кодирования («белый ящик»)
- Тестирование интеграций – проверка проектирования («черный ящик»)
- Тестирование правильности – проверка анализа требований («черный ящик»)
- Системное тестирование – проверка системного анализа
- Стрессовое тестирование
- Тестирование восстановления
- Тестирование производительности
- Тестирование безопасности

# Тестирование элементов

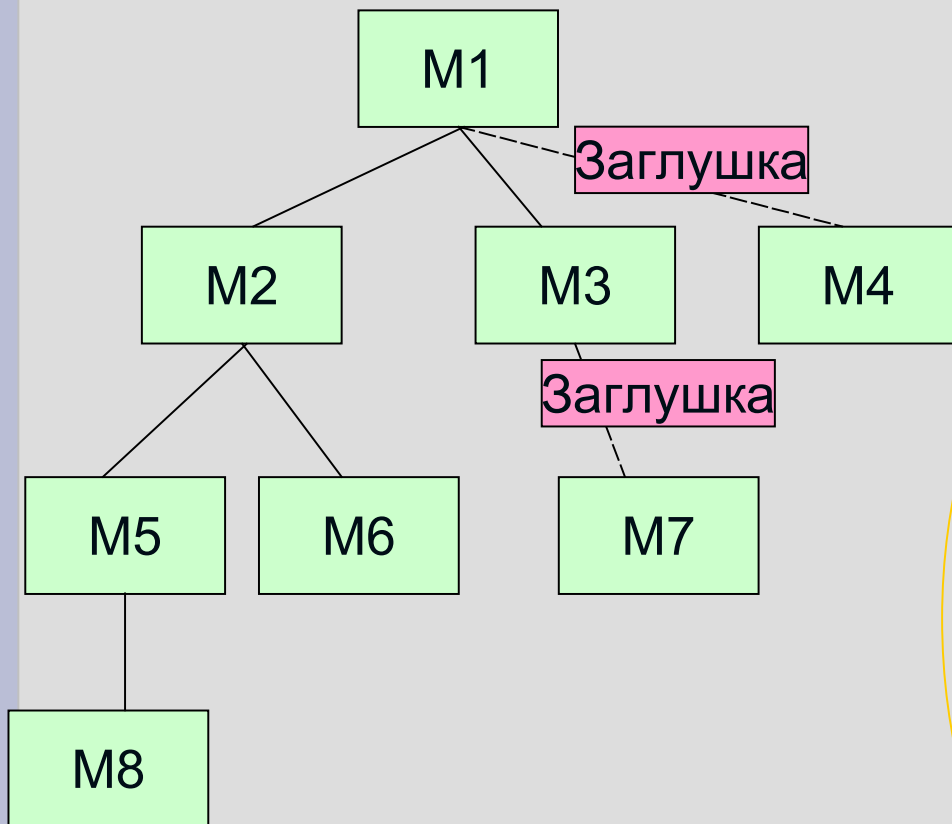
## Unit-тестирование

- Позволяет проверить:
  - Интерфейс модуля
  - Структуры данных
  - Независимые пути
  - Исключительные ситуации
  - Граничные условия
- Использует вспомогательные средства:
  - ◆ Драйвер – для запуска модуля
  - ◆ Заглушка – для замещения других модулей

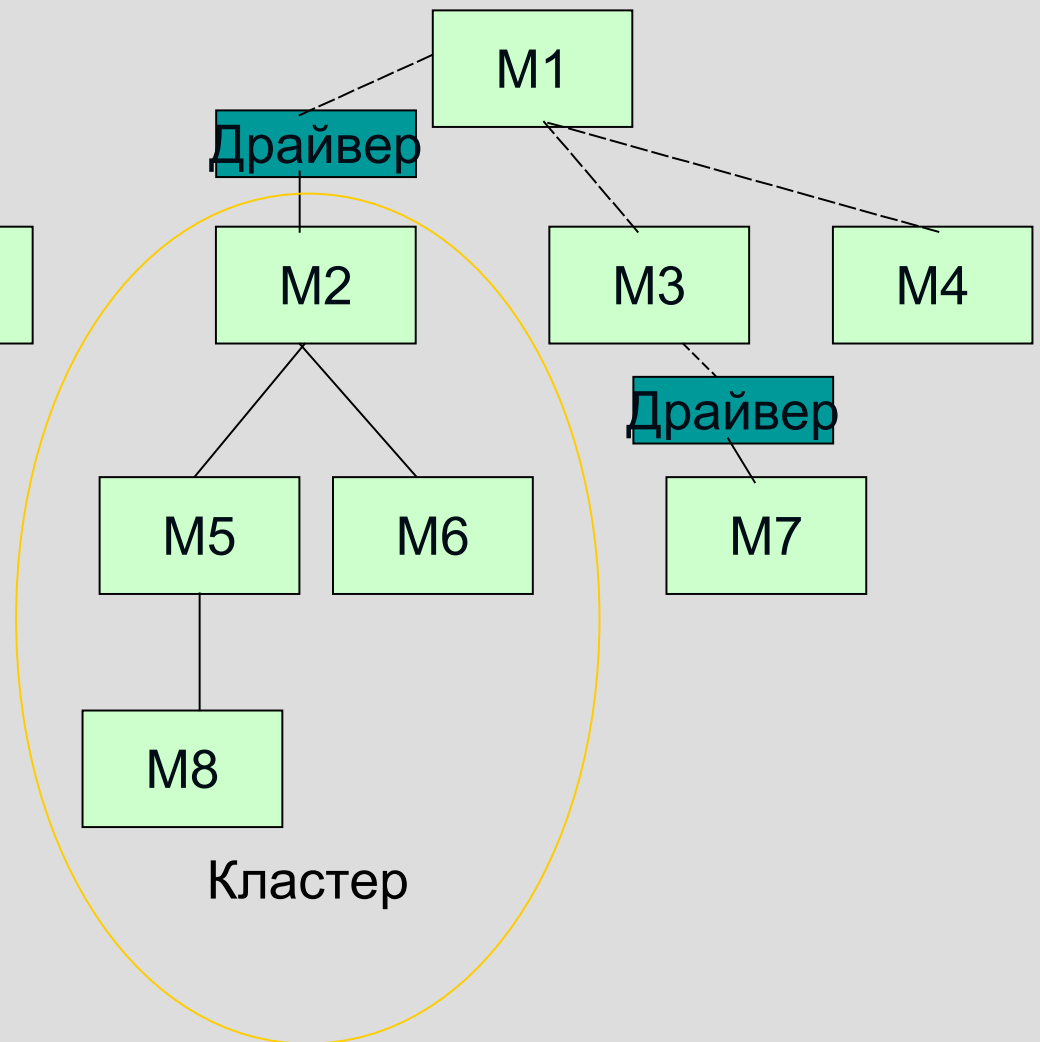
Программное обеспечение для Unit-тестирования: Google Test, Boost.

# Тестирование интеграции

Нисходящее



Восходящее



# Тестирование правильности

- Проверка соответствия функций ПО и документации (техническому заданию)
- Наличие и правильность оформления программной документации
  - Спецификации
  - Листинги
  - Руководства
  - Документы сопровождения (отчеты о проблемах)
- $\alpha$ -тестирование – проводит заказчик в организации разработчика
- $\beta$ -тестирование – проводит конечный пользователь в организации заказчика

# Системное тестирование

- Выявляет ошибки во взаимодействии программы и других элементов системы
- Проверка ввода от других элементов системы
- Проверка вывода (параллельный вывод на экран или печать)

# Стандарты на разработку программной документации

- Единая система программной документации (ЕСПД)
- ГОСТ 19.001-77 – ГОСТ 19.604-78 (28 ГОСТов)
- ГОСТ 19.701-90 (ИСО 5807-85) ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения
- [www.standards.ru](http://www.standards.ru), [protect.gost.ru](http://protect.gost.ru)

# Техническое задание

- ГОСТ 19.201-78. Техническое задание. Требования к содержанию и оформлению.
- Разделы ТЗ
  - Введение
  - Основания для разработки
  - Назначение разработки
  - Требования к программе
  - Требования к программной документации
  - Технико-экономические показатели
  - Стадии и этапы разработки
  - Порядок контроля и приемки
  - Приложения
- Существует утилита для автоматизации разработки технического задания

# Пояснительная записка

- Разделы:

- Введение

- Назначение и область применения

- Технические характеристики

- постановка задачи, описание применяемых математических методов и связанных с ними ограничений;
- описание алгоритма и (или) функционирования программы с обоснованием выбора, возможные взаимодействия программы с другими программами;
- описание и обоснование выбора метода организации входных и выходных данных;
- описание и обоснование выбора состава технических и программных средств, распределение носителей данных, которые использует программа.

- Ожидаемые технико-экономические показатели

- Источники, использованные при разработке

- Приложения (не обязательно)



# ГОСТ 19.101-77 Виды программ и программных документов

Компонент, комплекс  
Виды программных документов

Примеры документов:

Код вида документа	Вид документа
12	Текст программы
13	Описание программы
20	Ведомость эксплуатационных документов
32	Руководство системного программиста

# Эксплуатационные документы

- Ведомость эксплуатационных документов
- Руководство системного программиста
- Руководство программиста
- Руководство оператора (пользователя)
- Описание языка
- Руководство по техническому обслуживанию

# Обозначения программ и программных документов

- ГОСТ 19.103-77 Обозначения программ и программных документов
- Структура обозначения программ

А .	В .	XXXXXX -	XX
Код стран ы	Код организац ии- разработч ика	Регистрационный номер программного комплекса	Номер версии программы

- Структура обозначения документа

А .	В .	XXXXXX -	XX	XX	XX	- X
Код стран ы	Код организац ии- разработч ика	Регистрационный номер программного комплекса	Номер редакц ии докуме нта	Код вида докуме нта	Номер документ а данного вида	Номер части докумен та

# Спецификация

- ГОСТ 19.202-78 Спецификация. Требования к содержанию и оформлению
- Спецификация оформляется на листах формата А4 в форме таблицы:

<b>Обозначение</b>	<b>Наименование</b>	<b>Примечание</b>
--------------------	---------------------	-------------------

- Спецификация содержит разделы:
  - Документация
  - Комплексы
  - Компоненты

# Автоматизация процесса разработки документации

- Doxygen

- среда разработки (Code::Blocks)

- программный код с комментариями

- `double myFn(double x, double y)`
- `{`
- `/** \brief myFn - функция для вычисления суммы двух вещественных чисел, имеет два параметра:`
- `*`
- `* \param x - первое слагаемое типа double,`
- `* \param y - второе слагаемое типа double;`
- `* \return возвращает вещественное число, представляющее собой сумму двух слагаемых.`
- `*`
- `*/`
- `double z; /**< z - локальная переменная, используется для вычисления суммы`

# Фрагмент сгенерированного Докумен текста

The screenshot shows the LibreOffice Writer interface with a document titled 'refman.rtf'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Вставка', 'Формат', 'Стили', 'Таблица', 'Сервис', 'Окно', and 'Справка'. The toolbar contains various icons for file operations, editing, and formatting. The status bar at the bottom indicates 'Страница 4 из 5', '177 слов, 1 151 символов', 'Базовый', 'Русский', and 'TextSection'.

The document content is as follows:

double myFn (double x, double y)

myFn - функция для вычисления суммы двух вещественных чисел, имеет два параметра:

**Аргументы:**

<u>x</u>	- первое слагаемое типа <u>double</u> .
<u>y</u>	- второе слагаемое типа <u>double</u> .

**Возвращает:**

возвращает вещественное число, представляющее собой сумму двух слагаемых.

< z - локальная переменная, используется для вычисления суммы двух чисел

# Новые стандарты

- ГОСТ 19781-90 Обеспечение систем обработки информации программное. Термины и определения. Разработан взамен ГОСТ 19781-83 и ГОСТ 19.004-80 и устанавливает термины и определения понятий в области программного обеспечения (ПО) систем обработки данных (СОД), применяемые во всех видах документации и литературы, входящих в сферу работ по стандартизации или использующих результаты этих работ.
- ГОСТ 28388-89 Системы обработки информации. Документы на магнитных носителях данных. Порядок выполнения и обращения. Распространяется не только на программные, но и на конструкторские, технологические и другие проектные документы, выполняемые на магнитных носителях.

# Новые стандарты, основанные на международных стандартах

- ГОСТ Р ИСО/МЭК 9294-93 Информационная технология. Руководство по управлению документированием программного обеспечения. Стандарт полностью соответствует международному стандарту ИСО/МЭК ТО 9294:1990 и устанавливает рекомендации по эффективному управлению документированием ПС для руководителей, отвечающих за их создание.
- ГОСТ Р ИСО/МЭК 9126-93 Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. Стандарт полностью соответствует международному стандарту ИСО/МЭК 9126:1991. Стандарт определяет шесть комплексных характеристик, которые с минимальным дублированием описывают качество ПС (ПО, программной продукции): функциональные возможности; надежность; практичность; эффективность; сопровождаемость; мобильность. Эти характеристики образуют основу для дальнейшего уточнения и описания качества ПС.



- ГОСТ Р ИСО 9127-94 Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов. Стандарт полностью соответствует международному стандарту ИСО 9127:1989. Под документацией пользователя понимается документация, которая обеспечивает конечного пользователя информацией по установке и эксплуатации ПП. Под информацией на упаковке понимают информацию, воспроизводимую на внешней упаковке ПП.
- ГОСТ Р ИСО/МЭК 8631-94 Информационная технология. Программные конструктивы и условные обозначения для их представления. Описывает представление процедурных алгоритмов.

# Новые стандарты (продолжение)

- ГОСТ Р 51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию.
- ГОСТ Р 53622-2009 Информационная технология. Информационно-вычислительные системы. Стадии и этапы жизненного цикла, виды и комплектность документов.

# Системы управления версиями

## Термины

**SCM** (Software Configuration Management) — управление конфигурацией программного обеспечения.

**VCS** (Version Control System) — программное обеспечение, предназначенное для хранения нескольких версий элементов.

**Элемент** — файл, каталог, ссылки на файлы или каталоги.

**Репозиторий** — специальный каталог для хранения информации о версиях элементов.

**Ревизия** — текущее состояние репозитория.

# Классификация систем управления версиями

**По месту размещения оригиналов:**

централизованная

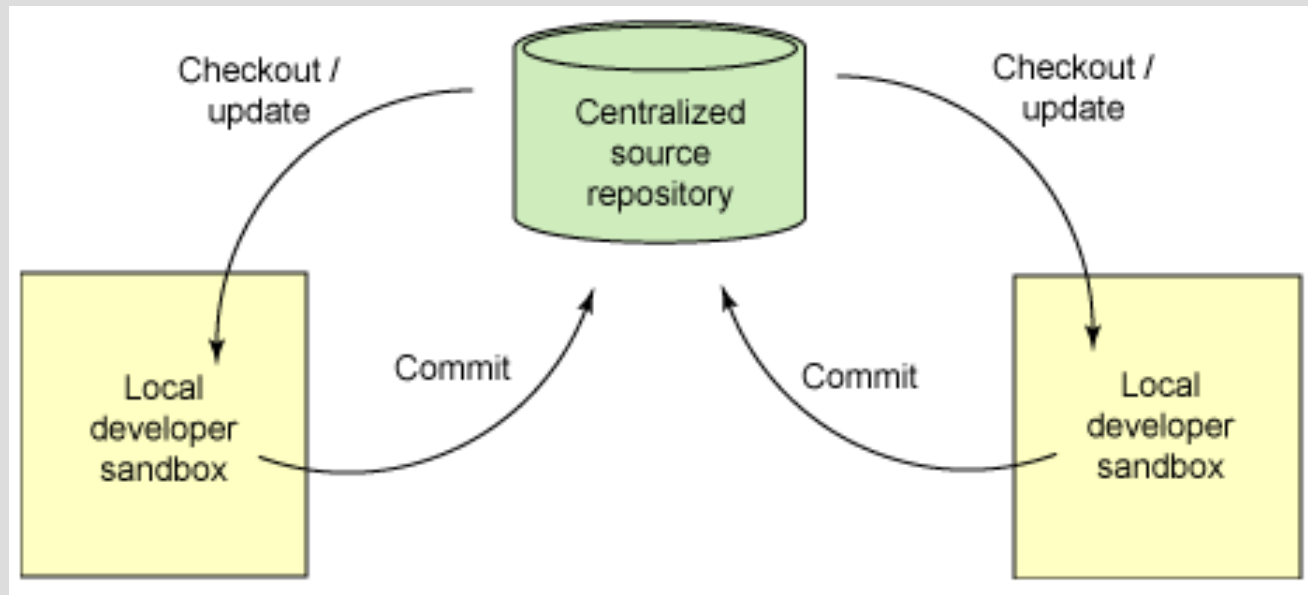
децентрализованная (распределенная)

**По способу хранения информации об  
изменениях:**

хранение версий элементов

хранение изменений

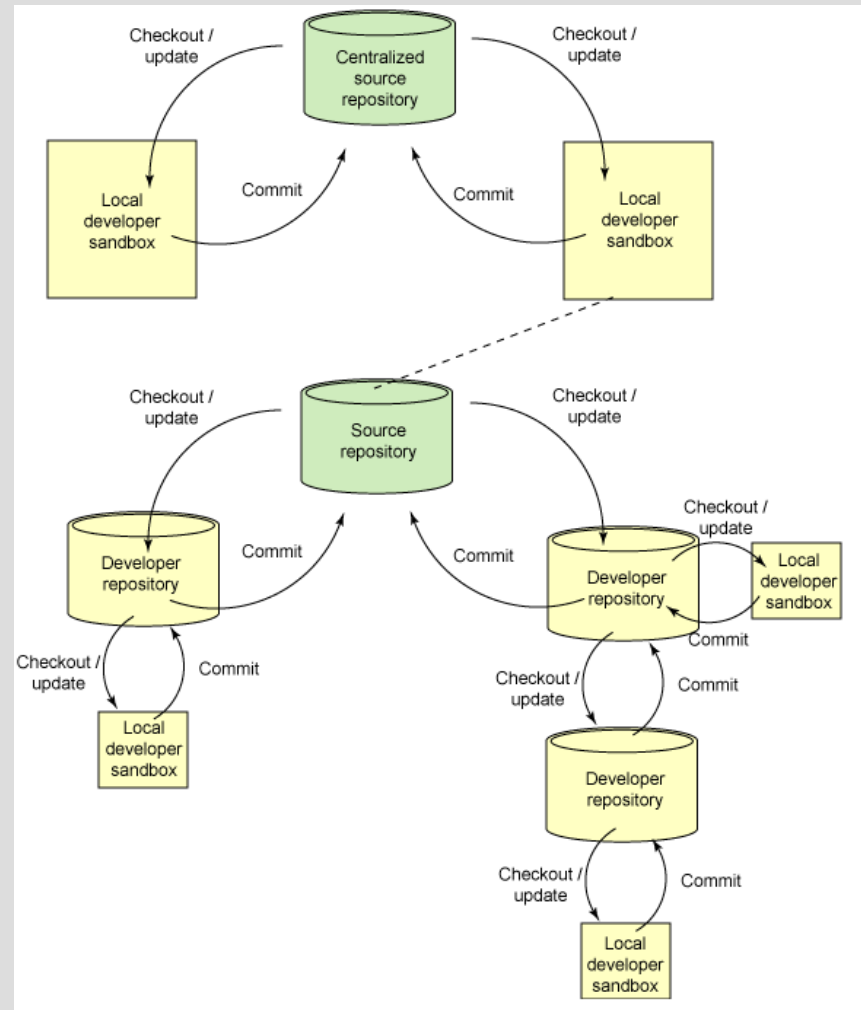
# Структура централизованной системы управления версиями



Статья:

[www.ibm.com](http://www.ibm.com) Version control for Linux, M.Tim Jones

# Структура децентрализованной системы управления версиями



Статья:

[www.ibm.com](http://www.ibm.com) Version control for Linux, M. Tim Jones

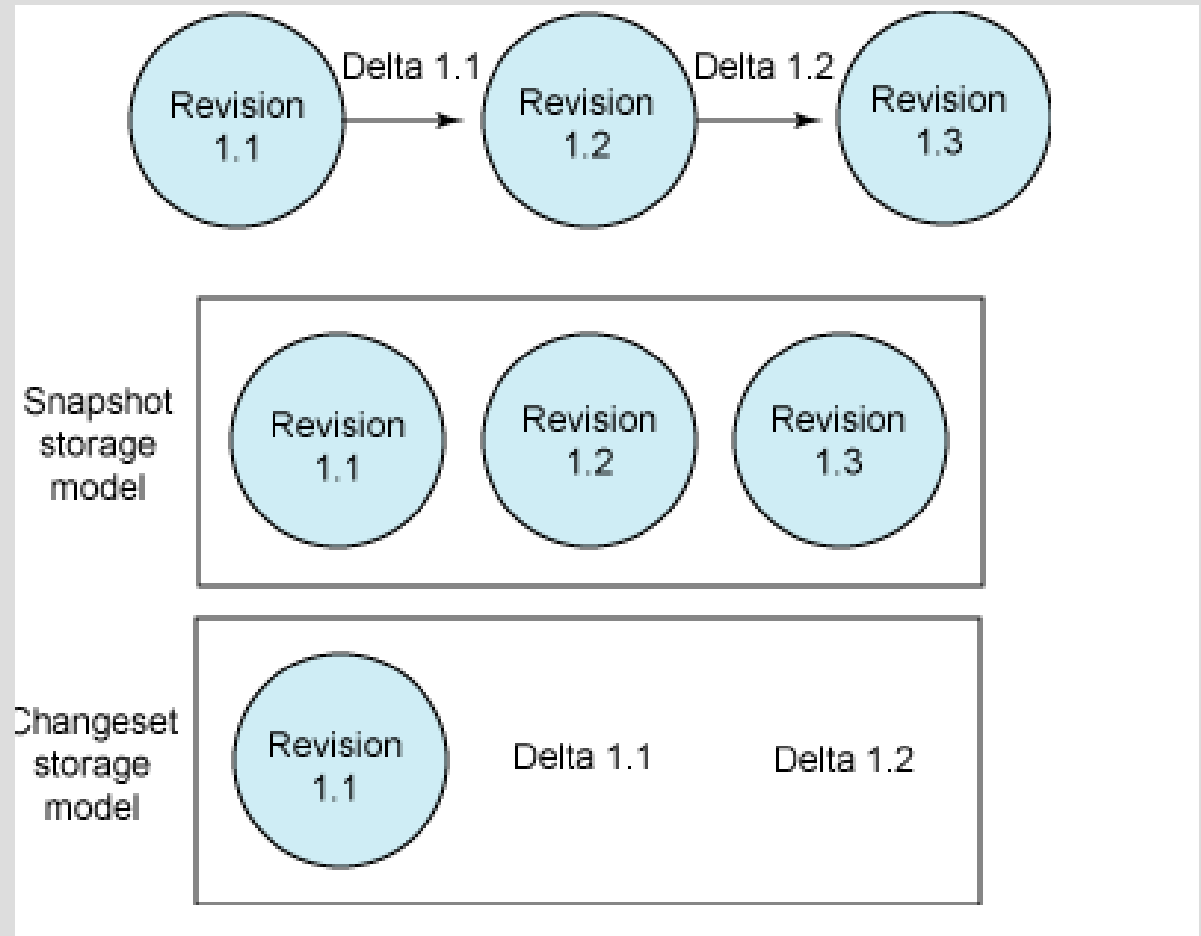
# Способы хранения информации об изменениях

Хранение версий файлов

Хранение изменений (приращений)

Статья:

[www.ibm.com](http://www.ibm.com) Version control for Linux, M.Tim Jones



# Примеры систем управления версиями

## Централизованные:

- CVS (Concurrent Versions System)
- Subversion (SVN)
  - TortoiseSVN
- ClearCase (IBM-Rational)
- Visual SourceSafe (Microsoft, до 2005 года)
- Team Foundation Server (Microsoft)

## Распределенные:

- Bazaar
- Mercurial
- Git



# Subversion

## Характеристика:

- Централизованная система
- Клиент-серверная архитектура
- Клиентское приложение: svn
- Протоколы для доступа к хранилищу
  - http
  - svn
- Собственная файловая система (FSFS)
- Хранение изменений файлов и каталогов
- Год выпуска первой версии — 2004
- Разработчик: CollabNet Inc.

# Структура каталогов репозитория

/rep

/ trunc

главная ветка разработки

/ tags

метки

/0.0.1

/0.0.2

...

/ branches

ответвления

/0.0.1

/0.0.2

...

# Структура хранилища Subversion. Ревизии

[\[править\]](#)

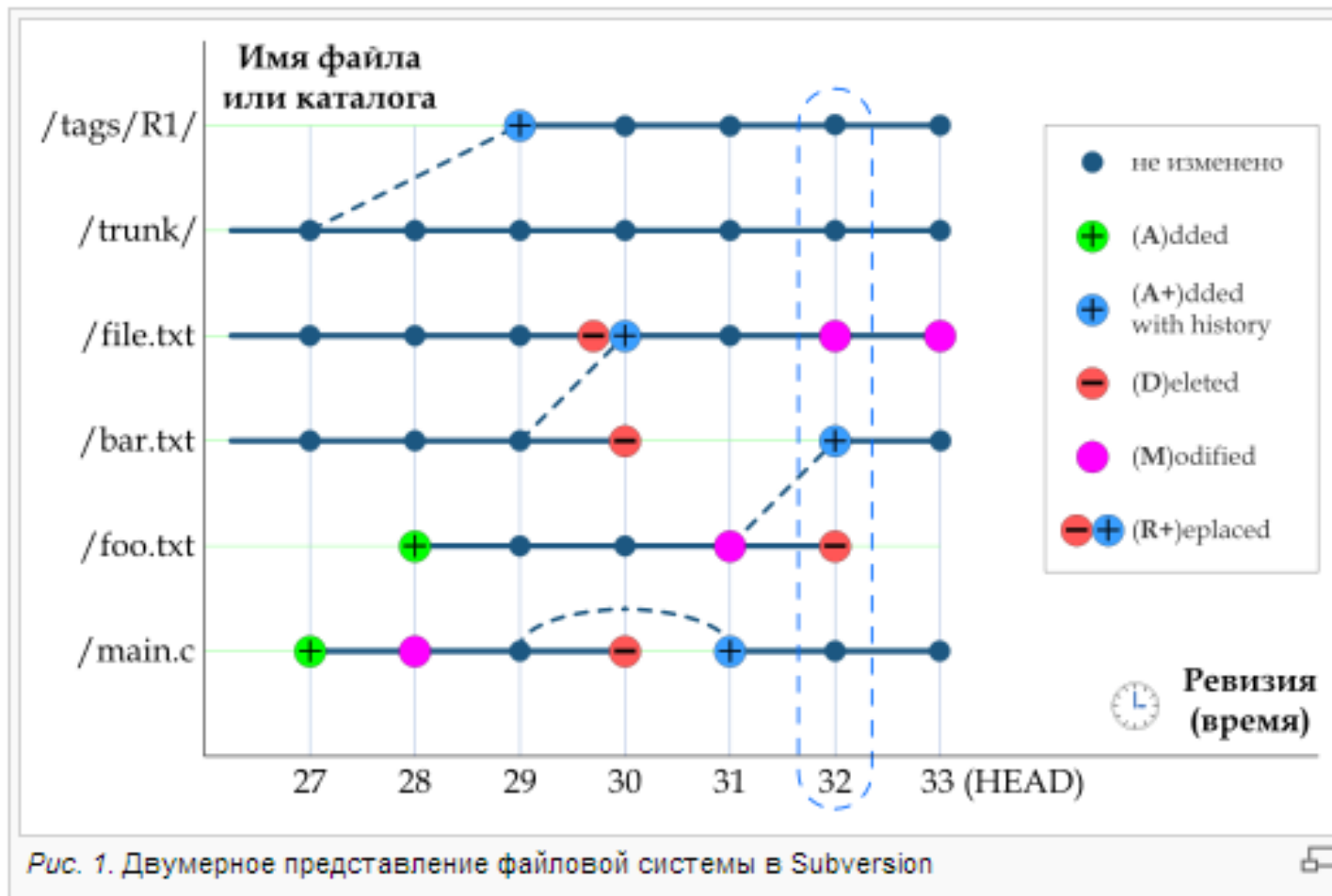


Рис. 1. Двумерное представление файловой системы в Subversion



# Команды Subversion

```
svn checkout svn://eniac/su91/4
```

```
svn update
```

```
svn ls <путь>
```

```
svn add ./my.txt
```

```
svn commit ./my.txt
```

```
svn status
```

```
svn log ./my.txt
```

```
svn diff ./my.txt
```

```
svn revert ./my.txt
```

```
svn merge ...
```

```
svn copy <откуда> <куда>
```

```
svn remove <путь/файл>
```

```
svn rename <путь/файл>
```

# Конфликты в репозитории

Причины:

рабочая копия репозитория (working copy) использует устаревшую ревизию, изменены файлы, которые вы не редактировали.

рабочая копия репозитория использует устаревшую ревизию, изменены файлы, которые редактировали вы,

некорректно был удален (перемещен, переименован) файл или каталог,

при слиянии ветвей указаны неправильные номера ревизий.