

Управление конфигурацией

Администрирование ИС

Лекция 9

Управление конфигурацией. Обзор метода

Управление конфигурациями определяется как процесс, с помощью которого администрация программы или проекта может *систематически идентифицировать, устанавливать связи, сопровождать и управлять различными компонентами программы или проекта.*

Этот процесс гарантирует *целостность компонент и трассируемость всех изменений*, возникающих в любой момент жизненного цикла программы или проекта.

Управление конфигурациями позволяет команде разработчиков программы или проекта точно *определять статус любой компоненты во все время ее жизненного цикла и позволяет перевоссоздать любую версию в любой момент времени.* Компонентами могут быть любые комбинации аппаратуры, программ, обслуживания и обучения.

Назначение метода

Метод управления конфигурациями обеспечивает порядок, в котором применяются техническое и административное руководство, и наблюдение.

Назначение методов:

1. Идентифицировать, определить и документировать объекты и схемы конфигураций
2. Прослеживать и управлять изменениями объектов и схем конфигураций
3. Запоминание и обработка изменений отчетов и статуса реализаций

Так как применение этого метода требует от команды разработчиков программы или проекта большого объема взаимосвязей, члены команды должны проявить *максимум общечеловеческих достоинств, таких как такт и способность договариваться*. В связи с этим *управление конфигурациями можно рассматривать как искусство и, точно так же, как техническую дисциплину*.

Этот метод предлагает достаточное количество деталей, которые полезны для больших, сложных программ или проектов.

Например, в качестве примера предлагается большое число трассировочных записей. При более ограниченных потребностях метод может быть адаптирован к более ограниченному варианту.

КЛЮЧЕВЫЕ ТЕРМИНЫ

Объектами конфигурации являются все основные результаты деятельности по разработке программы или проекта. Такие результаты идентифицируются и управляются с помощью управления конфигурациями.

Объекты конфигурации могут включать

- аппаратуру,
- программы,
- документацию,
- обучение и обслуживание.

Объектам конфигурации могут быть присвоены номера для простоты прослеживаний.

Заметим, что **термин "компонент"** используется в этом документе в очень обобщенном смысле, связываясь с некоторой частью общего решения. Термин "система" используется для обозначения полного решения, которое может включать программы, аппаратуру и другие компоненты.

КЛЮЧЕВЫЕ ТЕРМИНЫ

Схема - это "мгновенный снимок", замороженная конфигурация спецификаций или результатов, находящихся в разработке.

Схема может быть представлена *документом или набором документов*.

Создание схемы фактически представляет собой определение результирующего условия, возникающего при завершении каждого из основных шагов процесса разработки.

1. Когда процесс разработки переходит от одного шага к другому, специфические для этого шага результаты (документы, спецификации или продукты) пересматривают, чтобы убедиться в их качестве и связях (трассируемость) с предыдущими результатами.
2. Специфические версии объектов конфигурации, принадлежащие им, идентифицируются.
3. Когда (и если) результаты шага прошли процедуру обзора, они становятся схемой и теперь готовы к дальнейшей работе.
4. Эти схемы в дальнейшем подлежат формальному управлению изменениями, что позволяет гарантировать трассируемость объектов конфигурации так, чтобы изменениями можно было управлять адекватно.

КЛЮЧЕВЫЕ ТЕРМИНЫ

Процесс **управления изменениями** определяет оценку и трассировку запросов на изменения, анализ потенциального влияния изменений и принятие решений по внесению изменений в объекты конфигурации. Между управлением изменениями и управлением конфигурациями должен быть эффективный интерфейс. Реализация запросов на изменения должна включать трассировку данного изменения через процесс управления конфигурациями.

Версии и Редакции: Эти термины часто взаимозаменяемы. В данном документе термин "версия" используется прежде всего для ссылок на каждое уникальное появление объекта конфигурации, которому присвоен уникальный идентификационный номер. Редакцией здесь называется специфическая версия объекта конфигурации, предназначенная для "внешнего" использования. Редакциями будут версии объектов, образующие комплект системы для внутреннего тестирования или для передачи заказчику. Формальная версия-редакция называется просто "формальной редакцией".

Трассируемость называется возможность идентифицировать и историю, и текущее состояние (статус) каждого объекта конфигурации в любой точке жизненного цикла программы или проекта. Необходимой также является и возможность трассировки объектов конфигурации относительно требований заказчика, как они изложены в Спецификации Требований.

Гарантия качества включает в себя те процедуры и деятельности, которые позволяют гарантировать, что каждый объект будет реализован в ходе практической работы с требуемым качеством; гарантии основываются на верификации материалов, конструкции и постоянной проверке того, что объект удовлетворяет всем требованиям программы/проекта и подготовленным спецификациям.

НЕОБХОДИМОСТЬ В ДРУЖЕЛЮБИИ К ПОЛЬЗОВАТЕЛЯМ И ГИБКОСТИ

Система управления конфигурациями должна быть дружелюбна по отношению к пользователям; кроме того, она должна представлять собой простейшую систему, способную достичь требуемого уровня управления, требующую минимум документов и форм.

Система соответствует текущей работе, если она реализует корректное и адекватное управление, но не противостоит творческой свободе разработчиков.

Процесс управления внутренними конфигурациями должен взаимодействовать должным образом с аналогичными процессами партнеров по разработке, включенных в программу/проект.

Методология управления должна быть достаточно гибкой, чтобы обеспечить координацию этих (быть может, разнородных) процессов и определить все их последовательные интерфейсные точки.

ОТНОШЕНИЕ УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ К ДРУГИМ ПРОЦЕССАМ

Роль управления конфигурациями в рамках общей управленческой структуры программы/проекта **зависит прежде всего от общего объема работ.**

Управление конфигурациями тесно завязано с некоторыми другими процессами, включая *технология (собственно разработку), управление изменениями и гарантия качества (ГК).*

Этот процесс является *центральный* для управления всеми программными или проектными файлами, так как **он централизует управление изменениями в этих файлах.** Кроме того, он **гарантирует своевременность обмена информацией между всеми подразделениями организации.**

Взаимодействие управления конфигурациями и управления изменениями представляет собой хороший пример тесной связи между различными процессами и управлением конфигурациями.

1. Когда запрос на изменение попадает в процесс управления изменениями, об этом событии уведомляется менеджер конфигураций.
2. Это позволяет менеджеру конфигураций модифицировать запись, относящуюся к *Вычислению статусов конфигураций;*
3. затем объект от начала до конца прослеживается процессом управления конфигурации, пока не будет завершена его обработка процессом управления изменениями.
4. Разработчик завершает задачу, затребованную процессом управления изменениями, и затем уведомляет менеджер конфигураций об этом.
5. В заключение менеджер конфигураций передает изменение в ГК для последующей окончательной верификации.

В действительности существует неявный контакт между разработчиком и ГК в ходе внесения изменения.

МЕСТО УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ В ОРГАНИЗАЦИИ

Когда система управления конфигурациями установлена, первым вопросом в организации будет следующий: *"Где должно быть помещено управление конфигурации в нашей иерархии, чтобы был достигнут максимальный эффект?"*

Это место должно быть достаточно высоким в организационной иерархии, чтобы **обеспечить соответствующие возможности узнать и повлиять, и одновременно достаточно близким к собственно разработке, чтобы выполнять свои задачи.**

Слишком высокое место имеет склонность к созданию эффекта *"башни из слоновой кости"*: функция будет слишком удалена от производственного уровня.

Слишком низкое место может вызвать чрезмерные возможности наблюдения и влияния на ежедневную деятельность.

Ни одно из этих решений не является наилучшим; **для каждой организации должна быть сделана попытка приспособить базовые процессы к конкретным условиям.**

Например, если организация одновременно разрабатывает множество проектов или программ, она может создать в рамках своей административной структуры высокоуровневую функцию, ответственную за определение стандартного множества процедур и средств, которые затем приспособляются к специфическим потребностям программ или проектов. После этого каждый менеджер конфигурации будет нести ответственность за высокоуровневое управление конфигурацией своего проекта или программы, как это предусматривается установленными процедурами.

Заметим, что собственно организация должна оставаться внешней по отношению к команде разработчиков и технологической группе, так что персонал, относящийся к уровню организации, может сохранить независимые перспективы и осуществлять формальное управление над всеми процессами, проходящими в организации. Однако специфическая функция поддержки, подобная управлению программной конфигурацией, может быть передана технологической группе.

При меньших объемах разработок масштаб этой иерархии соответственно понижается; для очень маленьких или краткосрочных проектов менеджер проекта может выполнять все или большую часть функций управления конфигурациями.

ФУНКЦИИ УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ

Организация управления конфигурациями может быть поделена на несколько функций со специфическими обязанностями.

Одному человеку может поручено выполнение нескольких функций.

Персонал, ответственный за каждую функцию, разрабатывает и реализует линии поведения и процедуры для своих функций и гарантирует их выполнение.

Этими функциями являются:

- **Административный контроль и планирование конфигураций:** Ответственные за эту функцию специалисты обязаны обеспечить написание процедур, сопровождение разнообразных файлов управления конфигурациями и модификацию (по мере необходимости) записей ВСК.
- **Управление данными:** Эта функция управляет и отслеживает всю программную/проектную корреспонденцию и внешнюю документацию. Управление данными также контролирует все номера, присваиваемые документам, программам и аппаратуре.
- **Библиотеки:** В принципе могут быть выделены программные библиотекари, библиотекари документации и аппаратные библиотекари, в зависимости от типов объектов, определения которых должны быть запомнены в управляемой памяти. Нести ответственность за реализацию этой функции может как один человек, так и несколько.
- **Управление программными конфигурациями:** Эта функция реализует управление всеми разработанными кодами и всей документацией, связанной с разработкой программ.
- **Управление аппаратными конфигурациями:** Эта функция реализует управление всей аппаратурой. Последняя может включать покупное оборудование.

ПОДРАЗДЕЛЕНИЕ УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ

Обработка изменений является ключевым назначением внутреннего управления конфигурациями программы или проекта. Такая функция обычно поручается **Подразделению Управления Конфигурациями**, которое определяет схему результатов разработки и интегрирует все изменения к ней (после согласования всех последствий и анализа, проводимого процессом управления изменениями). Обязанности и организацию этого подразделения фиксируют в его уставе.

Подразделение Управления Конфигурациями может передать часть своих полномочий низкоуровневым подразделениям для обработки любого влияния на изменения внутренних схем и согласования воздействий исходных изменений.

Рассмотрим ряд возможных примеров подразделений, ответственных перед Подразделением Управления Конфигурациями:

- **Подразделение Управления Программными Конфигурациями или Подразделение Обзора Программных Конфигураций:** Обрабатывает все изменения, относящиеся к программам и обращенные к объектам внутренней схемы.
- **Подразделение Управления Аппаратными Конфигурациями:** Обрабатывает все изменения, относящиеся к покупному оборудованию и потенциально способные повлиять на программную или аппаратную разработку.
- **Подразделение Управления Интерфейсами:** Специализируется на обработке всех изменений как во внешних интерфейсах, так и во внутренних.

БИБЛИОТЕКИ УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ

Библиотеки управления конфигурациями используются для обеспечения гарантий по надлежащей идентификации, управлению, вычислению статуса, сохранности и восстановлению компьютерных программных продуктов, аппаратных технологических чертежей, документации, средств и прочих полезных/необходимых объектов.

Реально может быть использована любая комбинация (или интеграция) следующих четырех типов библиотек:

Библиотеки поддержки программных компонент: Эти библиотеки содержат поддерживающие материалы, используемые для помощи отдельным разработчикам программ при сохранении управления над результатами своей собственной работы без необходимости во внешнем управлении и деятельности, связанной с обработкой изменений. Магнитные ленты или какой-либо иной носитель со старыми версиями должны быть сохраняться, пока разработчик продолжает вносить изменения в программные компоненты или документы; в случаях необратимых неприятностей с системным диском, например, остается доступной вся исходная информация.

Головные библиотеки: Эти библиотеки находятся под жестким контролем со стороны управления конфигурациями. Они содержат все объекты, необходимые для управления хранением формально принятых программ и документов. Любая совокупность программных компонент и/или документов из такой библиотеки могут быть модифицированы только через обработку их формальным процессом управления изменениями.

Программные хранилища или архивы: Библиотеки этого типа являются общей памятью, в которой хранятся все объявленные созданными или закупленными программы и документы. Все программы и документация помещаются в такое хранилище для постоянного хранения в соответствие с условиями контракта или в соответствие с внутренними требованиями.

Технологические библиотеки: Библиотеки этого типа предназначены для сохранения всех средств, предусматриваемых применяемой технологией разработки. Объекты хранения включают средства для анализа риска, математические подпрограммы, программные модели, средства для оценки стоимости и другие программные средства, имеющие потенциальное значение для будущих проектов.

Процедура

Процесс управления конфигурациями состоит из следующих подпроцессов:

1. [Идентификация конфигураций](#)
2. [Контроль за конфигурациями](#)
3. [Вычисление статуса конфигурации](#)
4. [Аудиты/обзоры конфигураций](#)

Менеджеры программы или проекта должны планировать сроки и ресурсы для управления конфигурациями в самом начале жизненного цикла разрабатываемой системы.

Эта задача включает распределение ответственностей за каждую из работ, связанных с каждым из названных подпроцессов, и обеспечением гарантий того, что все предусматриваемые здесь работы определены и включены в План Управления Конфигурациями.

Рассматриваемые здесь различные процедуры на практике используются в разных формах, гибко, в зависимости от потребностей программы или проекта. Фундаментальным критерием применения процедур является следующее условие: *Может ли "судьба" каждого объекта конфигурации быть прослежена в течении всего жизненного цикла программы или проекта?*

1. ИДЕНТИФИКАЦИЯ КОНФИГУРАЦИЙ

Процесс идентификации конфигураций требует выполнения следующих шагов:

1. [Определение объектов конфигурации](#)
2. [Выбор схемы наименования объектов конфигурации](#)
3. [Определение утверждаемых схем](#)
4. [Определение внутренних схем](#)

ОПРЕДЕЛЕНИЕ ОБЪЕКТОВ КОНФИГУРАЦИИ

Каждый объект программы или проекта, который может изменять свое представление, содержание, структуру или статус, подпадает под управление конфигурациями и, таким образом, должен быть определен с самого начала программы или проекта. Такими объектами (и процессами для их определения) являются:

Корреспонденция: Команда разработчиков должна сохранять записи для прослеживания всей корреспонденции, хоть как-то связанной с разработкой, включая входящую и исходящую почту, телексы и факсы. Записи должны фиксировать любую связь между объектами (например, запрос-ответ, действие-реакция).

Версии документов: Обычно документы, создаваемые в ходе разработки программы или проекта, претерпевают ряд изменений, каждое из которых должно быть трассируемым и управляемым (**Предложения о работе или контракты, Спецификации, Поддерживающие руководства**).

Команды разработчиков, связанные с большими и сложными разработками и реализациями, обычно передают под контроль конфигурационного управления еще и проектные или программные планы, тестовые материалы, отчеты (обзорные, аудитные, проблемные), запросы на изменения, обучающие материалы, планы передач заказчику готовых материалов и пр.

Версии программных систем: Программные системы состоят из множества уровней. Множество подсистем может быть объединено в систему, а множество модулей - в подсистему. Любая комбинация этих объектов может образовать схему. После создания начальной схемы команда разработчиков создает первую из многих версий программы. Повторно используемые программные компоненты, образующие, как правило, отдельные слои, такие как системы управления базами данных, системы формирования отчетов, системы управления экранными формами и пр., не модифицируются, а обрабатывается (каждая поотдельности) как отдельная подсистема.

ОПРЕДЕЛЕНИЕ ОБЪЕКТОВ КОНФИГУРАЦИИ

Версии аппаратных компонент: Управление аппаратной конфигурацией аналогично управлению программной, рассмотренной в предыдущем разделе. Дополнительная проблема связана с прослеживанием аппаратной конфигурации, используемой при разработке программы. Изменение аппаратной компоненты может влиять на поведение программы, и наоборот, изменение в программе может вызвать изменение в поведении аппаратуры. Детальные записи вычислений статуса конфигурации необходимы для прослеживания этих изменений.

Планы аппаратных конфигураций: Команда разработчиков программы/проекта в любой момент процесса разработки может поместить рисунок, отражающий специфическую аппаратную конфигурацию, под конфигурационное управление.

Обучающие материалы: Обучающие материалы непосредственно связываются со схемой. Множество версий обучающих компонент часто бывают необходимы для согласования множества схем, зависящих от времени реализаций и плана реализации. Обучающие материалы включают много взаимозависимых компонент, таких как диалоговые инструкции, классы, видео- и аудио-ленты, компакт-диски и учебные программы. Каждая компонента обучающих материалов должна находиться под конфигурационным управлением как отдельный объект.

Обслуживающие компоненты: Обслуживающие компоненты включают все результаты лабораторных работ, которые заказчик оплачивает, но они не входят в какие-либо передаваемые ему программные/проектные объекты. Обучающие материалы могут рассматриваться как один из типов обслуживающих материалов. Однако для программ системной интеграции обучение играет настолько важную роль, что обучающие материалы предпочтительнее выделять из остальных обслуживающих материалов.

Поддерживающие элементы: К поддерживающим элементам относятся все объекты, которые входят в передаваемый заказчику комплект материалов, но не являются непосредственно программными или проектными. Примерами таких объектов являются небольшие программные средства и приспособления. Такие объекты, даже если они необходимы, могут не быть идентифицированы в спецификации передаваемых заказчику материалов.

ВЫБОР СХЕМЫ НАИМЕНОВАНИЯ ОБЪЕКТОВ КОНФИГУРАЦИИ

Каждому объекту конфигурации присваивает **идентификационный номер ID**.

Схема наименования, применяемая для идентификационных номеров, чаще всего включает следующие данные:

- *Тип объекта (например, документ или коды)*
- *Имя объекта*
- *Идентификация программы или проекта*
- *Номер версии*
- *Номер ревизии (ревизия для конкретной версии)*
- *Данные о готовности (информация, необходимая для кодирования готовности, может иметь вид, например, "R" - для состояния "Готово", и "N" - для состояния "Еще не готово")*

ОПРЕДЕЛЕНИЕ УТВЕРЖДАЕМЫХ СХЕМ

Менеджер программы или проекта определяет схемы для удовлетворения специфических потребностей. Следующие три схемы создаются в определенных точках жизненного цикла:

1. **Функциональная**, включает управляемые документы, такие как Функциональные Спецификации и Спецификации Приемо-Сдаточных Тестов.
2. **Размещения**, которая содержит Спецификацию Проекта Системы, тестовые планы и процедуры.
3. **Результаты**, которая содержит объекты конфигурации, образующие собственно систему (включая как разработанные, так и покупные компоненты).

ОПРЕДЕЛЕНИЕ ВНУТРЕННИХ СХЕМ

Во время любого процесса разработки менеджер программы/проекта может решиться на создание и использование **внутренней схемы**.

Такая схема обеспечивает моментальный снимок состояния разработки в некоторый момент времени.

Моменты времени задаются программным/проектным менеджером как точки инициализации контролирующих действий функции управления конфигурациями. Это обычно происходит, например, после того, как программный объект благополучно пройдет тестирование, но перед тем, как этот объект объявляется готовым к интеграции с другими объектами той же самой подсистемы. Такой точкой также является точка, в которой подразделения, подчиненные Подразделению Контроля за Конфигурациями, становятся активными.

2. КОНТРОЛЬ ЗА КОНФИГУРАЦИЯМИ

Управление конфигурациями непосредственно взаимодействует с процессом **управления изменений**. Первоначально этот шаг обеспечивает информацией управление изменениями о влиянии предлагаемого изменения.

Когда (и если) это изменение принимается, процесс управления конфигурациями гарантирует, что объекты конфигурации, на которые влияет данное изменение, будут пересмотрены, для чего он подготовит соответствующие идентификационные номера, а затем будет модифицирована и схема, на которую изменение оказало влияние.

Таким образом, управление конфигурациями обрабатывает объекты конфигурации и контролирует редакции и схемы. Процесс обработки изменений управляется Подразделением Контроля за Конфигурациями.

1. **УПРАВЛЕНИЕ ПЕРЕСМОТРЕННЫМИ ОБЪЕКТАМИ КОНФИГУРАЦИЯМИ**
2. **СИСТЕМЫ РЕГИСТРАЦИИ И ХРАНЕНИЯ**
3. **МОДИФИКАЦИЯ СХЕМ**

3. ВЫЧИСЛЕНИЕ СТАТУСА КОНФИГУРАЦИИ

После того, как изменение объекта конфигурации санкционировано, должна возникнуть некоторая временная задержка на время реализации изменения.

ВСК есть механизм, используемый для прослеживания эволюции каждого объекта системы и его текущего статуса, соотносящегося со спецификациями, обозначенными в общедоступной схемной документации и письменными соглашениями.

ВСК подчеркивают условие ко всему продукту в любой момент времени жизненного цикла его разработки и используются для контроля развития программы/проекта.

ВСК обеспечивает программного/проектного менеджера большим количеством информации о его продукте, включая то, как он разрабатывается и все ли требуемые свойства действительно реализованы.

В конечном счете, **ВСК обеспечивает** документацию о статусе каждого объекта конфигурации в любой момент процесса разработки.

Разработчики несут ответственность за состояние ВСК за счет своих действий, связанных с процессами управления конфигурациями. Средством, наиболее часто используемым в вычислении статусов, являются **базы данных трассируемости**, которые специально разрабатываются для прослеживания **следующих типов данных**:

- *Время возникновения каждой схемы и изменения*
- *Время определения каждого объекта конфигурации*
- *Описательная информация о каждом объекте конфигурации*
- *Статус запросов на изменения (принят, отклонен, ожидающий выполнения)*
- *Описание статусов*
- *Описательная информация о каждом запросе на изменение*
- *Статус изменения*
- *Описательная информация о каждом изменении*

4. АУДИТЫ И ОБЗОРЫ КОНФИГУРАЦИЙ

Менеджеры программы/проекта должны быть уверены, что требуемое управление конфигурацией реализуется - другими словами, все принятые изменения реализованы, а результат представляет собой то, что специфицировано в его проектной документации.

Для достижения необходимо высокого уровня доверия они планируют **регулярные аудиты и обзоры конфигурационного управления.**

Требования для этих аудитов и обзоров обычно специфицируются в **Плане Управления Конфигурациями**, но также они могут быть заданы в планах программы, проекта или качества, как это покажется подходящим.

Ответственность за нормальную реализацию аудитов конфигураций лежит на **менеджере конфигураций**, если эта должность предусмотрена, а если нет - на менеджере программы/проекта или менеджере качества.

Аудитам конфигураций адресуются следующие вопросы, относящиеся к измененным объектам конфигурации:

- *Проведены ли изменения так, как они специфицированы и проведены ли соответствующие технические обзоры?*
- *Выдержано ли следование соответствующим стандартам?*
- *Выдержано ли следование установленным процедурам управления конфигурациями для записи и выдачи отчетов?*
- *Модифицированы ли все связанные с изменением объекты конфигурации?*

Рынок систем конфигурационного управления

Без хорошего инструментария невозможно оперативно управлять конфигурациями ПО.

Специализированные системы конфигурационного управления позволяют выполнять автоматизированную сборку готового продукта из множества изменяющихся частей по типовой схеме, исключая "человеческий фактор" на этом рутинном этапе, а, следовательно, и дорогостоящие ошибки.

Можно выделить **четыре группы таких продуктов**:

- 1) обеспечивающие контроль версий (Rational ClearCase, Merant PVCS, Microsoft Visual SourceSafe);
- 2) обеспечивающие контроль версий и изменений (Rational ClearCase/ClearQuest, PVCS Professional);
- 3) обеспечивающие параллельную разработку, контроль версий, изменений и рабочих процессов (PVCS Dimensions, CCC:Harvest фирмы Computer Associates);
- 4) обеспечивающие все вышеуказанные возможности при взаимодействии нескольких географически удаленных команд (Rational MultiSite, PVCS Replicator).

Продукт ClearCase наиболее мощный и, соответственно, дорогой.

Он лучше всего подходит для проектов, в которых *участвует не менее 10 разработчиков и которые направлены на создание крупных систем или на выпуск тиражируемого ПО, когда особенно актуальны проблемы отслеживания версий.*

ClearCase поддерживает множество репозитариев проекта с их автоматической синхронизацией;

координирует совместную работу всех участников проекта;

управляет конфигурацией сборки ПО;

обеспечивает визуальный контроль над происходящим.

В числе **недостатков** ClearCase — необходимость серьезного администрирования, т.е. нагрузка при работе с продуктом перекладывается с конечных пользователей на администратора.

Продукт PVCS, встроен во многие системы разработки, предлагает *более простой версионный контроль*.

- Он хуже автоматизирует работу пользователей и сложнее в применении, но зато не требует отдельного сотрудника-администратора для поддержки небольших групп разработчиков.
- Существуют проблемы переходов от одного продукта PVCS к другому.
- Имеет меньшую стоимость.
- Может оказаться хорошим решением, если разработка идет под разными платформами (аппаратная платформа и ОС).

Продукт Microsoft Visual Source Safe осуществляет простой контроль исходных текстов и подходит для индивидуальной работы или для проектов, объединяющих нескольких человек. В нем нельзя организовать связь между участниками проекта, но он значительно дешевле и проще. Используется для ОС Windows 98, NT, 2000.

В ноябре 2002 г. компания Merant выпустила новую версию популярного инструмента для управления конфигурациями ПО **PVCS Professional 7.5**.

В состав пакета входят:

- PVCS Version Manager 7.5 — система контроля версий;
- PVCS Tracker Manager 7.5 — утилита формирования журнала изменений и задач;
- Configuration Builder 7.5 — утилита обеспечения стандартизированной и надежной компоновки готовых приложений.