

# Глава 6: Управление конфигурацией программного обеспечения

Из SWEBOK

## Содержание

- 1 Управление процессом SCM
  - 1.1 Организационный контекст SCM
  - 1.2 Ограничения и рекомендации для процесса SCM
  - 1.3 Планирование SCM
  - 1.4 План управления цепочками поставок
  - 1.5 Наблюдение за управлением конфигурацией программного обеспечения
- 2 Идентификация конфигурации программного обеспечения
  - 2.1 Определение элементов, подлежащих контролю
  - 2.2 Библиотека программного обеспечения
- 3 Управление конфигурацией программного обеспечения
  - 3.1 Запрос, оценка и утверждение изменений программного обеспечения
  - 3.2 Внесение изменений в программное обеспечение
  - 3.3 Отклонения и отказы
- 4 Учет состояния конфигурации программного обеспечения
  - 4.1 Информация о состоянии конфигурации программного обеспечения
  - 4.2 Отчет о состоянии конфигурации программного обеспечения
- 5 Аудит конфигурации программного обеспечения
  - 5.1 Аудит функциональной конфигурации программного обеспечения
  - 5.2 Аудит физической конфигурации программного обеспечения
  - 5.3 Текущий аудит базовой версии программного обеспечения
- 6 Управление выпуском программного обеспечения и его доставка
  - 6.1 Создание программного обеспечения
  - 6.2 Управление версиями программного обеспечения
- 7 инструментов управления конфигурацией программного обеспечения

## АКРОНИМЫ

<b>ЦКБ</b>	Плата управления конфигурацией
<b>СМ</b>	Управление конфигурацией
<b>FCA</b>	Аудит функциональной конфигурации
<b>СПС</b>	Аудит физической конфигурации
<b>SCCB</b>	Плата управления конфигурацией программного обеспечения
<b>SCI</b>	Элемент конфигурации программного обеспечения
<b>СКМ</b>	Управление конфигурацией программного обеспечения
<b>SCMP</b>	План управления конфигурацией программного обеспечения
<b>СКВ</b>	Запрос на изменение программного обеспечения
<b>SCSA</b>	Учет состояния конфигурации программного обеспечения
<b>SDD</b>	Документ по дизайну программного обеспечения
<b>СЭИ/СММИ</b>	Интеграция модели зрелости возможностей Института разработки программного обеспечения
<b>SQA</b>	Гарантия качества программного обеспечения
<b>СРС</b>	Спецификация требований к программному обеспечению

## ВВЕДЕНИЕ

Систему можно определить как комбинацию взаимодействующих элементов, организованных для достижения одной или нескольких заявленных целей [1]. Конфигурация системы — это функциональные и физические характеристики аппаратного или программного обеспечения, изложенные в технической документации или достигаемые в продукте [1]; его также можно рассматривать как набор определенных версий аппаратных средств, микропрограмм или программных элементов, объединенных в соответствии с определенными процедурами сборки для конкретной цели. Таким образом, управление конфигурацией (СМ) — это дисциплина идентификации конфигурации системы в определенные моменты времени с целью систематического контроля изменений в конфигурации и поддержания целостности и прослеживаемости конфигурации на протяжении всего жизненного цикла системы. Формально определяется как

Дисциплина, применяющая техническое и административное руководство и наблюдение для: идентификации и документирования функциональных и физических характеристик элемента конфигурации, управления изменениями этих характеристик, записи и отчета об обработке изменений и статусе реализации, а также проверки соответствия указанным требованиям. [1]

Управление конфигурацией программного обеспечения (SCM) — это процесс жизненного цикла вспомогательного программного обеспечения, который приносит пользу управлению проектами, деятельности по разработке и обслуживанию, деятельности по обеспечению качества, а также клиентам и пользователям конечного продукта. Концепции управления конфигурацией применяются ко всем элементам управления, хотя существуют некоторые различия в реализации между аппаратным СМ и программным СМ. SCM тесно связан с деятельностью по обеспечению качества программного обеспечения (SQA). Как определено в области знаний о качестве программного обеспечения (КА), процессы SQA обеспечивают уверенность в том, что программные продукты и процессы в жизненном цикле проекта соответствуют их заданным требованиям, путем планирования, принятия и выполнения набора действий для обеспечения адекватной уверенности в том, что качество соответствует требованиям. встраивается в программное обеспечение. Действия SCM помогают в достижении этих целей SQA. В некоторых контекстах проекта конкретные требования SQA предписывают определенные действия SCM.

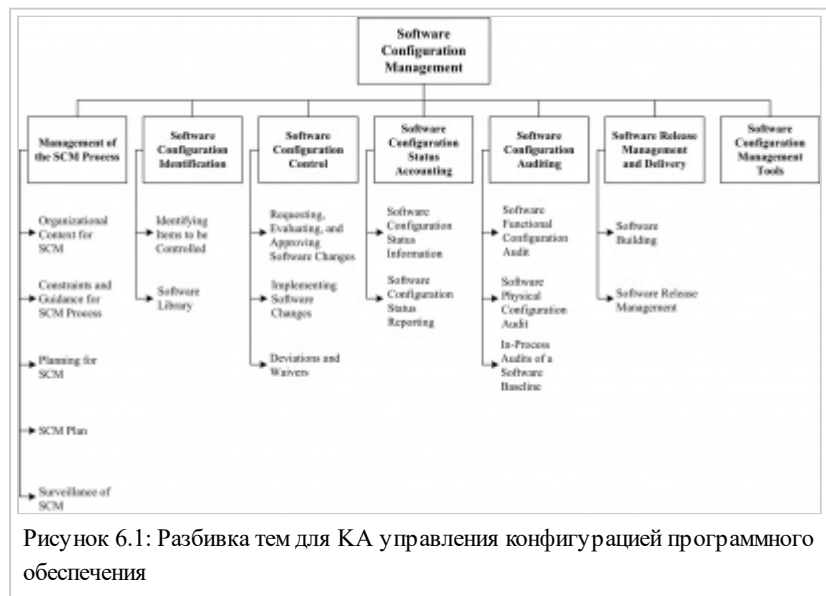
Действиями SCM являются управление и планирование процесса SCM, идентификация конфигурации программного обеспечения, контроль конфигурации программного обеспечения, учет состояния конфигурации программного обеспечения, аудит конфигурации программного обеспечения, а также управление выпусками программного обеспечения и их поставка. КА управления конфигурацией программного обеспечения связан со всеми другими КА, поскольку объектом управления конфигурацией является артефакт, создаваемый и используемый на протяжении всего процесса разработки программного обеспечения.

## РАЗБИВКА ТЕМ ПО УПРАВЛЕНИЮ КОНФИГУРАЦИЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Разбивка тем для КА управления конфигурацией программного обеспечения показана на рисунке 6.1.

### 1 Управление процессом SCM

SCM контролирует эволюцию и целостность продукта, идентифицируя его элементы; управление и контроль изменений; а также проверка, запись и отчетность по информации о конфигурации. С точки зрения инженера-программиста SCM облегчает разработку и внедрение изменений. Успешное внедрение SCM требует тщательного планирования и управления. Это, в свою очередь, требует понимания организационного контекста и ограничений, накладываемых на разработку и реализацию процесса SCM.



#### 1.1 Организационный контекст SCM

[ 2 , с6, прин. Г] [ 3 , введение][ 4 , с29]

Чтобы спланировать процесс SCM для проекта, необходимо понимать организационный контекст и отношения между организационными элементами. SCM взаимодействует с несколькими другими видами деятельности или организационными элементами. Организационные элементы, отвечающие за поддерживающие процессы разработки программного обеспечения, могут быть структурированы различными способами. Хотя ответственность за выполнение определенных задач SCM может быть

возложена на другие части организации (например, на организацию-разработчика), общая ответственность за SCM часто возлагается на отдельный организационный элемент или назначенное лицо. Программное обеспечение часто разрабатывается как часть более крупной системы, содержащей аппаратные и программные элементы. В таком случае, Действия SCM происходят параллельно с действиями CM аппаратного и микропрограммного обеспечения и должны быть согласованы с CM системного уровня. Обратите внимание, что прошивка содержит аппаратное и программное обеспечение; следовательно, применимы как аппаратные, так и программные концепции CM. SCM может взаимодействовать с деятельностью организации по обеспечению качества по таким вопросам, как управление записями и несоответствующие элементы. Что касается первого, то некоторые элементы, находящиеся под контролем SCM, могут также быть документами проекта, на которые распространяются положения программы обеспечения качества организации. Управление несоответствующими элементами обычно является обязанностью деятельности по обеспечению качества; однако SCM может помочь в отслеживании и составлении отчетов по элементам конфигурации программного обеспечения, попадающим в эту категорию. Возможно, самые тесные отношения связаны с организациями, занимающимися разработкой и обслуживанием программного обеспечения. Именно в этом контексте выполняются многие задачи по управлению конфигурацией программного обеспечения. Часто одни и те же инструменты поддерживают разработку, обслуживание и задачи SCM.

### 1.2 Ограничения и рекомендации для процесса SCM

[ 2 , с6, прин. Д, анн. Д ] [ 3 , с2, с5 ] [ 5 , с19с2.2 ]

Ограничения, влияющие на процесс SCM, и рекомендации для него исходят из ряда источников. Политики и процедуры, установленные на корпоративном или другом организационном уровне, могут влиять или предписывать разработку и реализацию процесса SCM для данного проекта. Кроме того, договор между покупателем и поставщиком может содержать положения, затрагивающие процесс SCM. Например, может потребоваться определенный аудит конфигурации или может быть указано, что определенные элементы должны быть помещены в CM. Когда разрабатываемые программные продукты потенциально могут повлиять на общественную безопасность, внешние регулирующие органы могут наложить ограничения. Наконец, конкретный процесс жизненного цикла программного обеспечения, выбранный для программного проекта, и уровень формализма, выбранный для реализации программного обеспечения, влияют на разработку и реализацию процесса SCM.

### 1.3 Планирование SCM

[ 2 , с6, прин. Д, анн. Д ] [ 3 , с23 ] [ 4 , с29 ]

Планирование процесса SCM для данного проекта должно соответствовать организационному контексту, применимым ограничениям, общепринятым руководствам и характеру проекта (например, размеру, критичности безопасности и безопасности). К основным видам деятельности относятся идентификация конфигурации программного обеспечения, управление конфигурацией программного обеспечения, учет состояния конфигурации программного обеспечения, аудит конфигурации программного обеспечения, а также управление выпусками программного обеспечения и их поставка. Кроме того, обычно рассматриваются такие вопросы, как организация и обязанности, ресурсы и графики, выбор и внедрение инструментов, контроль поставщиков и субподрядчиков, а также контроль интерфейса. Результаты деятельности по планированию записываются в план SCM (SCMP), который обычно подвергается анализу и аудиту SQA. Стратегии разветвления и слияния должны быть тщательно спланированы и доведены до сведения, поскольку они влияют на многие виды деятельности SCM. С точки зрения SCM ветвь определяется как набор развивающихся версий исходного файла [1]. Слияние заключается в объединении различных изменений в один и тот же файл [1]. Обычно это происходит, когда несколько человек изменяют элемент конфигурации. Существует множество широко используемых стратегий ветвления и слияния (дополнительное обсуждение см. в разделе «Дополнительная литература»). Модель жизненного цикла разработки программного обеспечения (см. Модели жизненного цикла программного обеспечения в КА процесса разработки программного обеспечения) также влияет на деятельность SCM, и планирование SCM должно учитывать это. Например, непрерывная интеграция является обычной практикой во многих подходах к разработке программного обеспечения. Обычно для него характерны частые циклы сборки-тестирования-развертывания.

#### 1.3.1 Организация SCM и обязанности

[ 2 , анн. Dс5, анн. Ds6 ] [ 3 , с10-11 ] [ 4 , введение, с29 ]

Чтобы избежать путаницы в отношении того, кто будет выполнять те или иные действия или задачи SCM, необходимо четко определить организационные роли, которые будут задействованы в процессе SCM. Конкретные обязанности для определенных действий или задач SCM также должны быть назначены организационным единицам либо по названию, либо по организационному элементу. Общие полномочия и каналы отчетности для SCM также должны быть определены, хотя это может быть достигнуто на этапе управления проектом или планирования обеспечения качества.

### 1.3.2 Ресурсы и расписания SCM

[ 2 , анн. Дс8] [ 3 , с23]

Планирование SCM определяет персонал и инструменты, участвующие в выполнении действий и задач SCM. Он решает вопросы планирования, устанавливая необходимую последовательность задач SCM и определяя их взаимосвязь с графиками и этапами проекта, установленными на этапе планирования управления проектом. Также указываются любые требования к обучению, необходимые для реализации планов и обучения новых сотрудников.

### 1.3.3 Выбор инструмента и его реализация

[ 3 , с26s2, с26s6] [ 4 , с29s5]

Как и в любой области разработки программного обеспечения, выбор и внедрение инструментов SCM должны быть тщательно спланированы. Следует рассмотреть следующие вопросы:

- Организация: что мотивирует приобретение инструментов с организационной точки зрения?
- Инструменты: можем ли мы использовать коммерческие инструменты или разработать их самостоятельно?
- Среда: какие ограничения накладывает организация и ее технический контекст?
- Наследие: как проекты будут использовать (или не использовать) новые инструменты?
- Финансирование: кто будет платить за инструменты, приобретение, обслуживание, обучение и настройку?
- Масштаб: как новые инструменты будут развернуты — например, во всей организации или только в конкретных проектах?
- Собственность: кто отвечает за внедрение новых инструментов?
- Будущее: каков план использования инструментов в будущем?
- Изменения: насколько адаптируемы инструменты?
- Ветвление и слияние: совместимы ли возможности инструментов с запланированными стратегиями ветвления и слияния?
- Интеграция: интегрируются ли между собой различные инструменты SCM? С другими инструментами, используемыми в организации?
- Миграция: можно ли портировать репозиторий, поддерживаемый инструментом контроля версий, в другой инструмент контроля версий, сохраняя при этом полную историю содержащихся в нем элементов конфигурации?

SCM обычно требует набора инструментов, а не одного инструмента. Такие наборы инструментов иногда называют верстаками. В таком контексте еще одним важным соображением при планировании выбора инструментов является определение того, будет ли рабочее место SCM открытым (другими словами, инструменты от разных поставщиков будут использоваться в разных действиях процесса SCM) или интегрированным (где элементы рабочего места предназначены для совместной работы). Размер организации и тип задействованных проектов также могут повлиять на выбор инструмента (см. раздел 7, Инструменты управления конфигурацией программного обеспечения).

### 1.3.4 Контроль поставщика/субподрядчика

[ 2 , с13] [ 3 , с13с9, с14с2]

Программный проект может приобретать или использовать приобретенные программные продукты, такие как компиляторы или другие инструменты. При планировании SCM учитывается, будут ли и каким образом эти элементы взяты под контроль конфигурации (например, интегрированы в библиотеки проекта), а также как будут оцениваться и управляться изменения или обновления. Аналогичные соображения применимы к программному обеспечению субподрядчиков. При использовании субподрядного программного обеспечения необходимо установить как требования SCM, которые должны

быть наложены на процесс SCM субподрядчика как часть субконтракта, так и средства контроля за соблюдением. Последнее включает в себя рассмотрение того, какая информация SCM должна быть доступна для эффективного мониторинга соответствия.

### 1.3.5 Управление интерфейсом

[ 2 , с12] [ 3 , с24с4]

Когда элемент программного обеспечения будет взаимодействовать с другим элементом программного или аппаратного обеспечения, изменение одного элемента может повлиять на другой элемент. При планировании процесса SCM учитывается, как будут идентифицированы взаимодействующие элементы и как будут управляться и сообщаться изменения в элементах. Роль SCM может быть частью более крупного системного процесса спецификации интерфейса и управления им; это может включать спецификации интерфейса, планы управления интерфейсом и документы по управлению интерфейсом. В этом случае планирование SCM для управления интерфейсом происходит в контексте процесса системного уровня.

### 1.4 План управления цепочками поставок

[ 2 , анн. Д] [ 3 , с23] [ 4 , с29с1]

Результаты планирования SCM для данного проекта записываются в план управления конфигурацией программного обеспечения (SCMP), «живой документ», который служит ссылкой для процесса SCM. Он поддерживается (то есть обновляется и утверждается) по мере необходимости в течение жизненного цикла программного обеспечения. При реализации SCMP обычно необходимо разработать ряд более подробных подчиненных процедур, определяющих, как конкретные требования будут выполняться в ходе повседневной деятельности, например, какие стратегии ветвления будут использоваться и как часто происходят сборки и выполняются автоматические тесты всех видов. Руководство по созданию и обслуживанию SCMP на основе информации, полученной в результате деятельности по планированию, доступно из ряда источников, таких как [2\*]. Эта ссылка содержит требования к информации, которая должна содержаться в SCMP;

- Введение (цель, объем, используемые термины)
- Управление SCM (организация, обязанности, полномочия, применимые политики, директивы и процедуры)
- Программный проект может приобретать или использовать действия SCM (идентификация конфигурации, управление конфигурацией и т. д.).
- Графики SCM (координация с другими мероприятиями проекта)
- Ресурсы SCM (инструменты, физические ресурсы и человеческие ресурсы)
- Техническое обслуживание SCMP.

### 1.5 Наблюдение за управлением конфигурацией программного обеспечения

[ 3 , с11с3]

После внедрения процесса SCM может потребоваться определенная степень надзора для обеспечения надлежащего выполнения положений SCMP. Скорее всего, будут особые требования SQA для обеспечения соответствия указанным процессам и процедурам SCM. Лицо, ответственное за SCM, следит за тем, чтобы лица, на которых возложена ответственность, правильно выполняли определенные задачи SCM. Орган по обеспечению качества программного обеспечения, как часть деятельности по аудиту соответствия, может также осуществлять это наблюдение. Использование интегрированных инструментов SCM с возможностью управления технологическим процессом может упростить задачу наблюдения. Некоторые инструменты облегчают соблюдение процессов, предоставляя инженеру-программисту гибкость в адаптации процедур. Другие инструменты навязывают процесс, оставляя инженеру-программисту меньше гибкости.

#### 1.5.1 Меры и измерения SCM

[ 3 , с9с2, с25с2–с3]

Меры SCM могут быть разработаны для предоставления конкретной информации о развивающемся продукте или для обеспечения понимания функционирования процесса SCM. Связанной с этим целью мониторинга процесса SCM является обнаружение возможностей для улучшения процесса. Измерения

процессов SCM обеспечивают хорошее средство для мониторинга эффективности деятельности SCM на постоянной основе. Эти измерения полезны для характеристики текущего состояния процесса, а также в качестве основы для проведения сравнений во времени. Анализ измерений может привести к пониманию, ведущему к изменениям процесса и соответствующим обновлениям SCMP. Библиотеки программного обеспечения и различные возможности инструментов SCM предоставляют источники для извлечения информации о характеристиках процесса SCM (а также для предоставления проектной и управленческой информации). Например, информация о времени, необходимом для внесения различных типов изменений, будет полезна при оценке критериев определения того, какие уровни полномочий являются оптимальными для авторизации определенных типов изменений и для оценки будущих изменений. Наблюдение за выводами, которые можно получить из измерений, а не за самими измерениями. Обсуждение процесса разработки программного обеспечения и измерения продукта представлено в КА процесса разработки программного обеспечения. Программы измерения программного обеспечения описаны в КА Управления разработки программного обеспечения. Необходимо позаботиться о том, чтобы сосредоточить внимание надзора на выводах, которые могут быть получены в результате измерений, а не на самих измерениях. Обсуждение процесса разработки программного обеспечения и измерения продукта представлено в КА процесса разработки программного обеспечения. Программы измерения программного обеспечения описаны в КА Управления разработки программного обеспечения. Необходимо позаботиться о том, чтобы сосредоточить внимание надзора на выводах, которые могут быть получены в результате измерений, а не на самих измерениях. Обсуждение процесса разработки программного обеспечения и измерения продукта представлено в КА процесса разработки программного обеспечения. Программы измерения программного обеспечения описаны в КА Управления разработки программного обеспечения.

### *1.5.2 Текущие аудиты SCM*

[ 3 , c1c1]

Аудиты могут проводиться в процессе разработки программного обеспечения для изучения текущего состояния конкретных элементов конфигурации или для оценки реализации процесса SCM. Внутривидовый аудит SCM обеспечивает более формальный механизм мониторинга отдельных аспектов процесса и может координироваться с функцией SQA (см. раздел 5, Аудит конфигурации программного обеспечения).

## **2 Идентификация конфигурации программного обеспечения**

[ 2 , c8] [ 4 , c29c1.1]

Идентификация конфигурации программного обеспечения идентифицирует элементы, подлежащие контролю, устанавливает схемы идентификации для элементов и их версий, а также устанавливает инструменты и методы, которые будут использоваться для получения и управления контролируемыми элементами. Эти действия обеспечивают основу для других действий SCM.

### *2.1 Определение элементов, подлежащих контролю*

[ 2 , c8s2.2] [ 4 , c29s1.1]

Одним из первых шагов в управлении изменениями является идентификация программных элементов, которыми нужно управлять. Это включает в себя понимание конфигурации программного обеспечения в контексте конфигурации системы, выбор элементов конфигурации программного обеспечения, разработку стратегии для маркировки элементов программного обеспечения и описания их взаимосвязей, а также определение как используемых базовых показателей, так и процедуры получения базовых показателей элементов.

#### *2.1.1 Конфигурация программного обеспечения*

[ 1 , c3]

Конфигурация программного обеспечения — это функциональные и физические характеристики аппаратного или программного обеспечения, указанные в технической документации или реализованные в продукте. Его можно рассматривать как часть общей конфигурации системы.

#### *2.1.2 Элемент конфигурации программного обеспечения*

[ 4 , c29s1.1]

Элемент конфигурации (CI) — это элемент или совокупность аппаратного или программного обеспечения или того и другого, предназначенные для управления как единым объектом. Элемент конфигурации программного обеспечения (SCI) — это программный объект, который был установлен как элемент конфигурации [1]. SCM обычно контролирует множество элементов в дополнение к самому коду. Элементы программного обеспечения, которые потенциально могут стать SCI, включают планы, спецификации и проектную документацию, материалы для тестирования, программные инструменты, исходный и исполняемый код, библиотеки кода, данные и словари данных, а также документацию по установке, обслуживанию, эксплуатации и использованию программного обеспечения. Выбор SCI — важный процесс, в котором должен быть достигнут баланс между обеспечением адекватной видимости для целей управления проектом и предоставлением управляемого количества контролируемых элементов.

### 2.1.3 Взаимосвязи элементов конфигурации программного обеспечения

[ 3 , с7с4]

Структурные отношения между выбранными SCI и их составными частями влияют на другие действия или задачи SCM, такие как создание программного обеспечения или анализ влияния предлагаемых изменений. Надлежащее отслеживание этих взаимосвязей также важно для поддержки прослеживаемости. При разработке схемы идентификации для SCI следует учитывать необходимость сопоставления идентифицированных элементов со структурой программного обеспечения, а также необходимость поддержки эволюции элементов программного обеспечения и их взаимосвязей.

### 2.1.4 Версия программного обеспечения

[ 1 , с3] [ 4 , с29с3]

Программные элементы развиваются по мере развития программного проекта. Версия элемента программного обеспечения — это идентифицированный экземпляр элемента. Его можно рассматривать как состояние развивающегося элемента. Вариант — это версия программы, полученная в результате применения разнообразия программного обеспечения.

### 2.1.5 Базовый уровень

[ 1 , с3]

Базовая версия программного обеспечения — это официально утвержденная версия элемента конфигурации (независимо от носителя), которая официально назначается и фиксируется в определенное время в течение жизненного цикла элемента конфигурации. Этот термин также используется для обозначения конкретной версии элемента конфигурации программного обеспечения, которая была согласована. В любом случае базовый план можно изменить только с помощью формальных процедур контроля изменений. Базовый план вместе со всеми утвержденными изменениями базового плана представляет собой текущую утвержденную конфигурацию. Обычно используемые базовые планы включают в себя функциональные, выделенные, развивающиеся и базовые планы продукта. Функциональный базовый план соответствует проверенным системным требованиям. Выделенный базовый уровень соответствует пересмотренной спецификации требований к программному обеспечению и спецификации требований к интерфейсу программного обеспечения. Базовый план разработки представляет собой изменяющуюся конфигурацию программного обеспечения в выбранные моменты жизненного цикла программного обеспечения. Полномочия на изменение этого базового состояния обычно принадлежат в первую очередь организации-разработчику, но могут быть переданы и другим организациям (например, SCM или Test). Базовый уровень продукта соответствует завершённому программному продукту, поставленному для системной интеграции. Базовые планы, которые будут использоваться для данного проекта, наряду с соответствующими уровнями полномочий, необходимыми для утверждения изменений, обычно определяются в SCMP. Базовый уровень продукта соответствует завершённому программному продукту, поставленному для системной интеграции. Базовые планы, которые будут использоваться для данного проекта, наряду с соответствующими уровнями полномочий, необходимыми для утверждения изменений, обычно определяются в SCMP.

### 2.1.6 Получение элементов конфигурации программного обеспечения

[ 3 , с18]



Элементы конфигурации программного обеспечения передаются под контроль SCM в разное время; то есть они включены в конкретную базовую линию в конкретный момент жизненного цикла программного обеспечения. Иницирующим событием является завершение некоторой формы формальной задачи приемки, такой как формальная проверка. Рисунок 6.2 характеризует рост базовых элементов по мере продолжения жизненного цикла. Этот рисунок основан на модели водопада только для иллюстрации; нижние индексы, используемые на рисунке, указывают версии эволюционирующих элементов. Запрос на изменение программного обеспечения (SCR) описан в разделе 3.1. При приобретении SCI необходимо установить его происхождение и первоначальную целостность. После приобретения SCI изменения в элементе должны быть официально одобрены в соответствии с SCI и задействованным базовым уровнем, как определено в SCMP. После утверждения,

## 2.2 Библиотека программного обеспечения

[ 3 , c1s3] [ 4 , c29s1.2]

Библиотека программного обеспечения — это контролируемая коллекция программного обеспечения и сопутствующей документации, предназначенная для помощи в разработке, использовании или обслуживании программного обеспечения [1]. Он также играет важную роль в управлении выпусками программного обеспечения и деятельности по доставке. Можно использовать несколько типов библиотек, каждая из которых соответствует определенному уровню зрелости элемента программного обеспечения. Например, рабочая

библиотека может поддерживать программирование, а библиотека поддержки проектов может поддерживать тестирование, а главная библиотека может использоваться для готовых продуктов. С каждой библиотекой связан соответствующий уровень управления SCM (соответствующий базовый уровень и уровень полномочий для внесения изменений). Безопасность с точки зрения контроля доступа и средств резервного копирования является ключевым аспектом управления библиотекой. Инструмент(ы), используемые для каждой библиотеки, должны поддерживать потребности управления SCM для этой библиотеки — как с точки зрения управления SCI, так и с точки зрения управления доступом к библиотеке. На уровне рабочей библиотеки это возможность управления кодом, предназначенная для разработчиков, специалистов по сопровождению и SCM. Он ориентирован на управление версиями элементов программного обеспечения при поддержке деятельности нескольких разработчиков. На более высоких уровнях управления доступ более ограничен, и основным пользователем является SCM. Эти библиотеки также являются важным источником информации для измерения работы и прогресса. доступ более ограничен, и SCM является основным пользователем. Эти библиотеки также являются важным источником информации для измерения работы и прогресса. доступ более ограничен, и SCM является основным пользователем. Эти библиотеки также являются важным источником информации для измерения работы и прогресса.

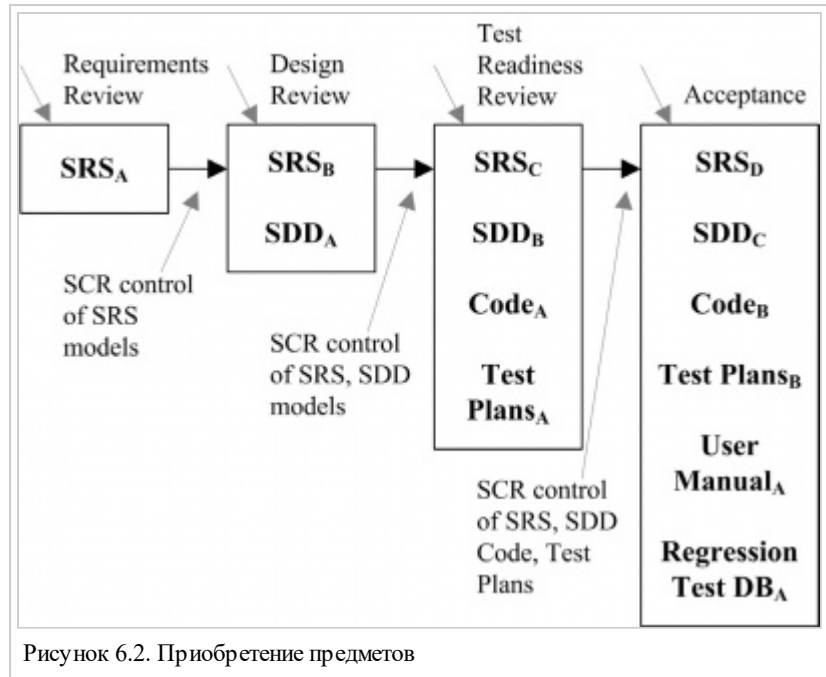


Рисунок 6.2. Приобретение предметов

## 3 Управление конфигурацией программного обеспечения

[ 2 , c9] [ 4 , c29c2]

Управление конфигурацией программного обеспечения связано с управлением изменениями в течение жизненного цикла программного обеспечения. Он охватывает процесс определения того, какие изменения необходимо внести, полномочия по утверждению определенных изменений, поддержку реализации этих изменений и концепцию формальных отклонений от требований проекта, а также отказ от них. Информация, полученная в результате этих действий, полезна для измерения потока изменений и поломок, а также аспектов доработки.

### 3.1 Запрос, оценка и утверждение изменений программного обеспечения

[ 2 , c9s2.4] [ 4 , c29s2]

Первым шагом в управлении изменениями контролируемых элементов является определение того, какие изменения необходимо внести. Процесс запроса на изменение программного обеспечения (см. типичную последовательность процесса запроса на изменение на рис. 6.3) обеспечивает формальные процедуры для подачи и регистрации запросов на изменение, оценки потенциальной стоимости и влияния предлагаемого изменения, а также принятия, модификации, отсрочки или отклонения изменений. предложенное изменение. Запрос на изменение (CR) — это запрос на расширение или уменьшение объема проекта; изменять политики, процессы, планы или процедуры; изменять расходы или бюджеты; или пересмотреть графики [1]. Запросы на изменение элементов конфигурации программного обеспечения могут исходить от кого угодно в любой момент жизненного цикла программного обеспечения и могут включать предлагаемое решение и запрошенный приоритет. Одним из источников CR является инициирование корректирующих действий в ответ на сообщения о проблемах. Независимо от источника тип изменения (например, дефект или улучшение) обычно записывается в CR программного обеспечения (SCR). Это дает возможность отслеживать дефекты и собирать данные об изменениях по типам изменений. После получения SCR выполняется техническая оценка (также известная как анализ воздействия) для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. тип изменения (например, дефект или усовершенствование) обычно записывается в CR программного обеспечения (SCR). Это дает возможность отслеживать дефекты и собирать данные об изменениях по типам изменений. После получения SCR выполняется техническая оценка (также известная как анализ воздействия) для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. тип изменения (например, дефект или усовершенствование) обычно записывается в CR программного обеспечения (SCR). Это дает возможность отслеживать дефекты и собирать данные об изменениях по типам изменений. После получения SCR выполняется техническая оценка (также известная как анализ воздействия) для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. дефект или усовершенствование) обычно записывается в CR программного обеспечения (SCR). Это дает возможность отслеживать дефекты и собирать данные об изменениях по типам изменений. После получения SCR выполняется техническая оценка (также известная как анализ воздействия) для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. тип изменения (например, дефект или усовершенствование) обычно записывается в CR программного обеспечения (SCR). Это дает возможность отслеживать дефекты и собирать данные об изменениях по типам изменений. После получения SCR выполняется техническая оценка (также известная как анализ воздействия) для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. Это дает возможность отслеживать дефекты и собирать данные об изменениях по типам изменений. После получения SCR выполняется техническая оценка (также известная как анализ воздействия) для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение.

элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. Это дает возможность отслеживать дефекты и собирать данные об изменениях по типам изменений. После получения SCR выполняется техническая оценка (также известная как анализ воздействия) для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. техническая оценка (также известная как анализ воздействия) выполняется для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение. техническая оценка (также известная как анализ воздействия) выполняется для определения объема модификаций, которые потребуются, если запрос на изменение будет принят. Для этой задачи важно хорошее понимание взаимосвязей между программными (и, возможно, аппаратными) элементами. Наконец, установленный орган — в соответствии с затронутым базовым состоянием, вовлеченным SCI и характером изменения — оценит технические и управленческие аспекты запроса на изменение и либо примет, либо модифицирует, либо отклонит, либо отложит предложенное изменение.

### 3.1.1 Плата управления конфигурацией программного обеспечения

[ 2 , c9s2.2] [ 3 , c11s1p1p2]

Полномочия на принятие или отклонение предлагаемых изменений возложены на организацию, обычно известную как Совет по контролю конфигурации (CCB). В небольших проектах эти полномочия могут фактически принадлежать лидеру или назначенному лицу, а не совету, состоящему из нескольких человек. Может быть несколько уровней полномочий на изменение в зависимости от множества критериев, таких как

критичность задействованного элемента, характер изменения (например, влияние на бюджет и график) или текущий момент жизненного цикла проекта. Состав CCB, используемых для данной системы, варьируется в зависимости от этих критериев (представитель SCM всегда присутствует). Представлены все заинтересованные стороны, соответствующие уровню CCB. Когда объем полномочий CCB строго касается программного обеспечения, он известен как Совет по контролю конфигурации программного обеспечения (SCCB).

### 3.1.2 Процесс запроса на изменение программного обеспечения

[ 3 , c1c4, c8c4]

Эффективный процесс запроса на изменение программного обеспечения (SCR) требует использования вспомогательных инструментов и процедур для создания запросов на изменение, обеспечения выполнения процесса изменения, регистрации решений CCB и предоставления информации о процессе изменения. Связь между этой возможностью инструмента и системой сообщения о проблемах может облегчить отслеживание решений для заявленных проблем.

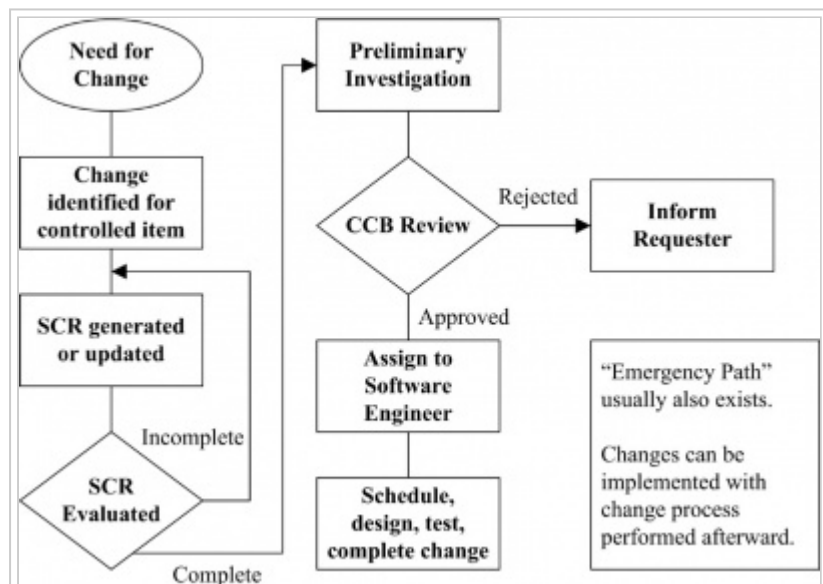


Рисунок 6.3. Поток процесса управления изменениями

### 3.2 Внесение изменений в программное обеспечение

[ 4 , с29]

Утвержденные SCR реализуются с использованием определенных программных процедур в соответствии с применимыми требованиями графика. Поскольку несколько утвержденных SCR могут быть реализованы одновременно, необходимо предоставить средства для отслеживания того, какие SCR включены в конкретные версии и базовые версии программного обеспечения. В рамках завершения процесса внесения изменений завершённые изменения могут пройти аудит конфигурации и проверку качества программного обеспечения, включая проверку того, что были внесены только утверждённые изменения. Процесс запроса на изменение программного обеспечения, описанный выше, обычно документирует информацию об утверждении SCM (и другой информации) для изменения. Изменения могут поддерживаться средствами контроля версий исходного кода. Эти инструменты позволяют группе инженеров-программистов или одному инженеру-программисту отслеживать и документировать изменения в исходном коде. Эти инструменты предоставляют единый репозиторий для хранения исходного кода, могут предотвратить одновременное редактирование одного и того же модуля более чем одним инженером-программистом и записывают все изменения, внесенные в исходный код. Инженеры-программисты проверяют модули из репозитория, вносят изменения, документируют изменения, а затем сохраняют отредактированные модули в репозитории. При необходимости изменения также можно отменить, восстановив предыдущий базовый уровень. Более мощные инструменты могут поддерживать параллельную разработку и географически распределенные среды. Эти инструменты могут быть реализованы в виде отдельных специализированных приложений под управлением независимой группы SCM. Они также могут появиться как интегрированная часть среды разработки программного обеспечения. Наконец, они могут быть такими же элементарными, как рудиментарная система управления изменениями, поставляемая с операционной системой. может предотвратить одновременное редактирование одного и того же модуля более чем одним инженером-программистом и записывать все изменения, внесенные в исходный код. Инженеры-программисты проверяют модули из репозитория, вносят изменения, документируют изменения, а затем сохраняют отредактированные модули в репозитории. При необходимости изменения также можно отменить, восстановив предыдущий базовый уровень. Более мощные инструменты могут поддерживать параллельную разработку и географически распределенные среды. Эти инструменты могут быть реализованы в виде отдельных специализированных приложений под управлением независимой группы SCM. Они также могут появиться как интегрированная часть среды разработки программного обеспечения. Наконец, они могут быть такими же элементарными, как рудиментарная система управления изменениями, поставляемая с операционной системой. и запишите все изменения, внесенные в исходный код. Инженеры-программисты проверяют модули из репозитория, вносят изменения, документируют изменения, а затем сохраняют отредактированные модули в репозитории. При необходимости изменения также можно отменить, восстановив предыдущий базовый уровень. Более мощные инструменты могут поддерживать параллельную разработку и географически распределенные среды. Эти инструменты могут быть реализованы в виде отдельных специализированных приложений под управлением независимой группы SCM. Они также могут появиться как интегрированная часть среды разработки программного обеспечения. Наконец, они могут быть такими же элементарными, как рудиментарная система управления изменениями, поставляемая с операционной системой. и запишите все изменения, внесенные в исходный код. Инженеры-программисты проверяют модули из репозитория, вносят изменения, документируют изменения, а затем сохраняют отредактированные модули в репозитории. При необходимости изменения также можно отменить, восстановив предыдущий базовый уровень. Более мощные инструменты могут поддерживать параллельную разработку и географически распределенные среды. Эти инструменты могут быть реализованы в виде отдельных специализированных приложений под управлением независимой группы SCM. Они также могут появиться как интегрированная часть среды разработки программного обеспечения. Наконец, они могут быть такими же элементарными, как рудиментарная система управления изменениями, поставляемая с операционной системой. изменения также можно отменить, восстановив

предыдущий базовый уровень. Более мощные инструменты могут поддерживать параллельную разработку и географически распределенные среды. Эти инструменты могут быть реализованы в виде отдельных специализированных приложений под управлением независимой группы SCM. Они также могут появиться как интегрированная часть среды разработки программного обеспечения. Наконец, они могут быть такими же элементарными, как рудиментарная система управления изменениями, поставляемая с операционной системой. Изменения также можно отменить, восстановив предыдущий базовый уровень. Более мощные инструменты могут поддерживать параллельную разработку и географически распределенные среды. Эти инструменты могут быть реализованы в виде отдельных специализированных приложений под управлением независимой группы SCM. Они также могут появиться как интегрированная часть среды разработки программного обеспечения. Наконец, они могут быть такими же элементарными, как рудиментарная система управления изменениями, поставляемая с операционной системой.

### 3.3 Отклонения и отказы

[ 1 , с3]

Ограничения, наложенные на усилия по разработке программного обеспечения, или спецификации, созданные во время деятельности по разработке, могут содержать положения, которые не могут быть выполнены в назначенный момент жизненного цикла. Отклонение — это письменное разрешение, выданное до изготовления изделия, на отклонение от конкретных требований к характеристикам или конструкции для определенного количества единиц или определенного периода времени. Отказ от требований — это письменное разрешение на принятие элемента конфигурации или другого назначенного элемента, который во время производства или после представления на проверку отклоняется от установленных требований, но, тем не менее, считается пригодным для использования как есть или после доработки утвержденным методом. В этих случаях используется формальный процесс для получения одобрения отклонений или отказов от положений.

## 4 Учет состояния конфигурации программного обеспечения

[ 2 , с10]

Учет состояния конфигурации программного обеспечения (SCSA) — это элемент управления конфигурацией, состоящий из регистрации и предоставления информации, необходимой для эффективного управления конфигурацией.

### 4.1 Информация о состоянии конфигурации программного обеспечения

[ 2 , с10s2.1]

Деятельность SCSA разрабатывает и управляет системой сбора и представления необходимой информации по мере прохождения жизненного цикла. Как и в любой информационной системе, информация о состоянии конфигурации, которой необходимо управлять для меняющихся конфигураций, должна быть идентифицирована, собрана и сохранена. Различная информация и измерения необходимы для поддержки процесса SCM и удовлетворения потребностей в отчетах о состоянии конфигурации для управления, разработки программного обеспечения и других связанных действий. Типы доступной информации включают идентификацию утвержденной конфигурации, а также идентификацию и текущий статус реализации изменений, отклонений и отказов. Некоторая форма поддержки автоматизированных инструментов необходима для выполнения задач сбора данных и отчетности SCSA; это может быть возможность базы данных, автономный инструмент,

### 4.2 Отчет о состоянии конфигурации программного обеспечения

[ 2 , с10s2.4] [ 3 , с1s5, с9s1, с17]

Отчетная информация может использоваться различными организационными и проектными элементами, включая группу разработчиков, группу обслуживания, управление проектом и деятельность по обеспечению качества программного обеспечения. Отчетность может принимать форму специальных запросов для ответов на определенные вопросы или периодического создания предварительно разработанных отчетов. Некоторая информация, полученная в результате деятельности по учету состояния в ходе жизненного цикла, может стать записями обеспечения качества. В дополнение к отчету о текущем состоянии конфигурации информация, полученная SCSA, может служить основой для различных измерений. Примеры включают количество запросов на изменение на SCI и среднее время, необходимое для реализации запроса на изменение.

## 5 Аудит конфигурации программного обеспечения

[ 2 , с11]

Аудит программного обеспечения — это независимая проверка рабочего продукта или набора рабочих продуктов для оценки соответствия спецификациям, стандартам, договорным соглашениям или другим критериям [1]. Аудиты проводятся в соответствии с четко определенным процессом, состоящим из различных ролей и обязанностей аудитора. Следовательно, каждый аудит должен быть тщательно спланирован. Аудит может потребовать от нескольких лиц выполнения различных задач в течение довольно короткого периода времени. Инструменты для поддержки планирования и проведения аудита могут значительно облегчить этот процесс. Аудит конфигурации программного обеспечения определяет степень, в которой элемент удовлетворяет требуемым функциональным и физическим характеристикам. Неформальные аудиты такого типа могут проводиться в ключевые моменты жизненного цикла. В соответствии с регулирующим договором могут потребоваться два типа официальных аудитов (например, в контрактах, касающихся критически важного программного обеспечения): Аудит функциональной конфигурации (FCA) и Аудит физической конфигурации (PCA). Успешное завершение этих аудитов может быть предпосылкой для установления базовой линии продукта.

### 5.1 Аудит функциональной конфигурации программного обеспечения

[ 2 , с11s2.1]

Целью FCA программного обеспечения является обеспечение того, чтобы проверяемый элемент программного обеспечения соответствовал его руководящим спецификациям. Результаты деятельности по проверке и проверке программного обеспечения (см. Проверка и проверка в КА качества программного обеспечения) являются ключевыми входными данными для этого аудита.

### 5.2 Аудит физической конфигурации программного обеспечения

[ 2 , с11s2.2]

Цель аудита физической конфигурации программного обеспечения (PCA) состоит в том, чтобы убедиться, что проектная и справочная документация соответствует готовому программному продукту.

### 5.3 Текущий аудит базовой версии программного обеспечения

[ 2 , с11s2.3]

Как упоминалось выше, аудиты могут проводиться в процессе разработки для изучения текущего состояния конкретных элементов конфигурации. В этом случае аудит может быть применен к выборочным базовым элементам, чтобы убедиться, что производительность соответствует спецификациям, или чтобы убедиться, что разрабатываемая документация продолжает соответствовать разрабатываемому базовому элементу.

## 6 Управление выпуском программного обеспечения и его доставка

[ 2 , с14] [ 3 , с8с2]

В этом контексте выпуск относится к распространению элемента конфигурации программного обеспечения вне деятельности по разработке; это включает внутренние выпуски, а также распространение среди клиентов. Когда для доставки доступны разные версии элемента программного обеспечения (например, версии для разных платформ или версии с различными возможностями), часто необходимо воссоздать определенные версии и упаковать правильные материалы для доставки версии. Библиотека программного обеспечения является ключевым элементом в выполнении задач по выпуску и доставке.

### 6.1 Создание программного обеспечения

[ 4 , с29s4]

Создание программного обеспечения — это деятельность по объединению правильных версий элементов конфигурации программного обеспечения с использованием соответствующих данных конфигурации в исполняемую программу для доставки заказчику или другому получателю, например, деятельность по тестированию. Для систем с аппаратным обеспечением или микропрограммой исполняемая программа доставляется в действие по сборке системы. Инструкции по сборке гарантируют, что необходимые шаги сборки выполняются в правильной последовательности. В дополнение к созданию программного обеспечения для новых выпусков, SCM обычно также необходимо иметь возможность воспроизводить

предыдущие выпуски для восстановления, тестирования, обслуживания или дополнительных выпусков. Программное обеспечение создается с использованием определенных версий вспомогательных инструментов, таких как компиляторы (см. Основы компиляции в Computing Foundations KA). Возможно, потребуется воссоздать точную копию ранее созданного элемента конфигурации программного обеспечения. В этом случае вспомогательные инструменты и соответствующие инструкции по сборке должны находиться под контролем SCM, чтобы обеспечить доступность правильных версий инструментов. Возможности инструмента полезны для выбора правильных версий элементов программного обеспечения для заданной целевой среды и для автоматизации процесса сборки программного обеспечения из выбранных версий и соответствующих данных конфигурации. Для проектов с параллельными или распределенными средами разработки эта возможность инструмента необходима. Большинство сред разработки программного обеспечения предоставляют эту возможность. Эти инструменты различаются по сложности: от требования, чтобы инженер-программист изучил специализированный язык сценариев, до подходов, ориентированных на графику, которые скрывают большую часть сложности «интеллектуального» средства сборки. Процесс сборки и продукты часто подлежат проверке качества программного обеспечения. Выходные данные процесса сборки могут понадобиться для дальнейшего использования и могут стать записями по обеспечению качества.

## 6.2 Управление версиями программного обеспечения

[ 4 , c29s3.2]

Управление выпуском программного обеспечения включает в себя идентификацию, упаковку и доставку элементов продукта — например, исполняемой программы, документации, примечаний к выпуску и данных конфигурации. Учитывая, что изменения в продукте могут происходить на постоянной основе, одной из проблем управления выпусками является определение времени выпуска выпуска. Серьезность проблем, решаемых в выпуске, и измерения плотности ошибок предыдущих выпусков влияют на это решение. Задача упаковки должна определить, какие элементы продукта должны быть доставлены, а затем выбрать правильные варианты этих элементов с учетом предполагаемого применения продукта. Информация, документирующая физическое содержание выпуска, известна как документ с описанием версии. В примечаниях к выпуску обычно описываются новые возможности, известные проблемы, и требования к платформе, необходимые для правильной работы продукта. Выпускаемый пакет также содержит инструкции по установке или обновлению. Последнее может быть осложнено тем фактом, что у некоторых текущих пользователей могут быть версии, устаревающие на несколько выпусков. В некоторых случаях управление выпуском может потребоваться для отслеживания распространения продукта среди различных клиентов или целевых систем — например, в случае, когда поставщик должен уведомить клиента о новых обнаруженных проблемах. Наконец, можно реализовать механизм обеспечения целостности выпущенного элемента, например, путем выпуска вместе с ним цифровой подписи. Для поддержки этих функций управления выпусками необходимы возможности инструмента. Полезно иметь соединение с функцией инструмента, поддерживающей процесс запроса на изменение, чтобы сопоставить содержимое выпуска с полученными SCR. Эта возможность инструмента может также поддерживать информацию о различных целевых платформах и различных клиентских средах.

## 7 инструментов управления конфигурацией программного обеспечения

[ 3 , c26s1] [ 4 , c8s2]

При обсуждении инструментов управления конфигурацией программного обеспечения полезно классифицировать их. Инструменты SCM можно разделить на три класса в зависимости от области, в которой они обеспечивают поддержку: индивидуальная поддержка, поддержка, связанная с проектом, и поддержка процесса в масштабах всей компании.

*Индивидуальные инструменты поддержки* подходят и обычно достаточны для небольших организаций или групп разработчиков без вариантов их программных продуктов или других сложных требований SCM. Они включают:

- Инструменты контроля версий: отслеживайте, документируйте и храните отдельные элементы конфигурации, такие как исходный код и внешнюю документацию.
- Инструменты обработки сборки: в простейшей форме такие инструменты компилируют и связывают исполняемую версию программного обеспечения. Более продвинутые инструменты сборки извлекают последнюю версию из программного обеспечения контроля версий, выполняют

проверки качества, запускают регрессионные тесты и создают различные формы отчетов, среди прочих задач.

- Инструменты управления изменениями: в основном поддерживают управление запросами на изменение и уведомлениями о событиях (например, изменения состояния запроса на изменение, достигнутые вехи).

*Инструменты поддержки, связанные с проектом, в основном поддерживают управление рабочим пространством для групп разработчиков и интеграторов; обычно они могут поддерживать распределенные среды разработки. Такие инструменты подходят для средних и крупных организаций с вариантами своих программных продуктов и параллельной разработкой, но без требований сертификации.*

*Инструменты поддержки общекорпоративных процессов обычно могут автоматизировать части общекорпоративного процесса, обеспечивая поддержку управления рабочими процессами, ролями и обязанностями. Они могут обрабатывать множество элементов, данных и жизненных циклов. Такие инструменты дополняют поддержку, связанную с проектом, поддерживая более формальный процесс разработки, включая требования сертификации.*

## ДАЛЬНЕЙШИЕ ЧТЕНИЯ

Стивен П. Берчук и Брэд Эшплтон, *Шаблоны управления конфигурацией программного обеспечения: эффективная командная работа, практическая интеграция* [6].

В этой книге полезные практики и стратегии SCM представлены в виде шаблонов. Шаблоны могут быть реализованы с использованием различных инструментов, но они выражаются независимым от инструментов способом.

«СММІ для разработки», версия 1.3, стр. 137–147 [7].

Эта модель представляет собой набор лучших практик, которые помогут организациям, занимающимся разработкой программного обеспечения, улучшить свои процессы. На уровне зрелости 2 он предлагает действия по управлению конфигурацией.

## ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

- [1] *ISO/IEC/IEEE 24765:2010 Системная и программная инженерия. Словарь*, ISO/IEC/IEEE, 2010.
- [2] *Стандарт IEEE. 828-2012, Стандарт управления конфигурацией в системной и программной инженерии*, IEEE, 2012.
- [3] AMJ Nass, *Принципы и практика управления конфигурацией*, 1-е изд., Addison Wesley, 2003.
- [4] И. Соммервиль, *Программная инженерия, 9-е изд.*, 9-е изд., Аддисон Уэсли, 2011.
- [5] Дж. В. Мур, *Дорожная карта разработки программного обеспечения : руководство , основанное на стандартах*, издательство Wiley-IEEE Computer Society Press, 2006.
- [6] С.П. Берчук и Б. Эшплтон, *Шаблоны управления конфигурацией программного обеспечения: эффективная командная работа, практическая интеграция*, Addison Wesley Professional, 2003.
- [7] Команда разработчиков СММІ, «СММІ для разработки, версия 1.3», Институт программной инженерии, 2010 г., <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=9661>.

Получено с " [http://swbokwiki.org/index.php?title=Chapter\\_6:\\_Software\\_Configuration\\_Management&oldid=602](http://swbokwiki.org/index.php?title=Chapter_6:_Software_Configuration_Management&oldid=602) "

- 
- Последнее изменение этой страницы состоялось 26 августа 2015 г., в 11:10.