Программная инженерия

Конфигурационное управление (Software Configuration Management)

Глава базируется на IEEE Guide to the Software Engineering Body of Knowledge - SWEBOK®, 2004. Содержит перевод описания области знаний SWEBOK® “Software Configuration Management”, с комментариями и замечаниями.

Официальный сайт “Основ программной инженерии” (по SWEBOK) - http://swebok.sorlik.ru

Программная инженерия

Конфигурационное управление (Software Configuration Management)

Система может быть определена как коллекция компонент, организованных для выполнения заданных функций или реализации комплекса функциональности (IEEE 610.12-90, Standard Glossary for Software Engineering Terminology). *Конфигурация* системы – функциональные и/или физические характеристики аппаратного, программно-аппаратного, программного обеспечения или их комбинации, сформулированные в технической документации и реализованные в продукте. Конфигурация также может восприниматься как сочетание конкретных версий аппаратных, программно-аппаратных или программных элементов, объединенных вместе, в соответствии с заданными процедурами сборки и отвечающих определенному назначению. *Конфигурационное управление* (CM - Configuration Management), в свою очередь, дисциплина идентификации конфигурации системы в определенные (заданные) моменты времени, с целью систематического контроля изменений конфигурации, а также поддержки и сопровождения целостной и отслеживаемой (трассируемой) конфигурации на протяжении всего жизненного цикла системы.

Конфигурационное управление формально определяется глоссарием IEEE 610 как “дисциплина приложения технических и административных указаний (инструкций) и контроля (надзора) для: идентификации и документирования функциональных и физических характеристик элементов конфигураций, контроля (управления) изменений этих характеристик, записи (сохранения) и ведения отчетности по обработке изменений и статусу их реализации, а также проверки (верификации) соответствия заданным требованиям.”

В соответствии с ГОСТ Р ИСО/МЭК (ISO/IEC, IEEE) 12207, *конфигурационное управление в области программного обеспечения* (“6.2 Управление конфигурацией” по ГОСТ) – *Software Configuration Management (SCM*)* – один из вспомогательных процессов жизненного цикла по стандарту 12207, поддерживающих проектный менеджмент, деятельность по разработке и сопровождению, обеспечению качества, а также, заказчиков и пользователей конечного продукта.

* в ряде источников можно увидеть аббревиатуру SCCM – Software Configuration and Change Management. При том, что в понимании SWEBOK и соответствующих стандартов, содержание SCM и SCCM тождественно, термин SCCM иногда используется для того, чтобы подчеркнуть принципиальную значимость управления изменениями как составной части конфигурационного управления.

Концепции конфигурационного управления применяются в отношении всех элементов, которые необходимо контролировать (несмотря на то, что существуют определенные отличия между конфигурационным управлением в приложении к аппаратному и программному обеспечению).

SCM-деятельность тесно связана с работами по обеспечению качества программного обеспечения (*Software Quality Assurance - SQA*). В соответствии с определением области знаний SWEBOK “Качество программного обеспечения” (Software Quality), SQA-процессы обеспечивают гарантию того, что программные продукты и процессы жизненного цикла в проекте соответствуют заданным требованиям, за счет планирования и выполнения работ, направленных на достижение определенного (приемлемого, *прим. автора*) уровня качества создаваемого программного продукта. SCM-деятельность помогает в достижении этих SQA-целей. В контексте некоторых проектов, определенные работы по конфигурационному управлению задаются требованиями SQA (например, в IEEE 730-02 “Standard for Software Quality Assurance Plans”).

Работы по конфигурационному управлению <программного обеспечения> включают: управление и планирование SCM-процессов, идентификацию программных конфигураций, контроль конфигураций, учет статусов конфигураций, аудит, а также управление выпуском (release management) и поставкой (delivery).

На рисунке 1 изображено стилизованное представление этих работ.

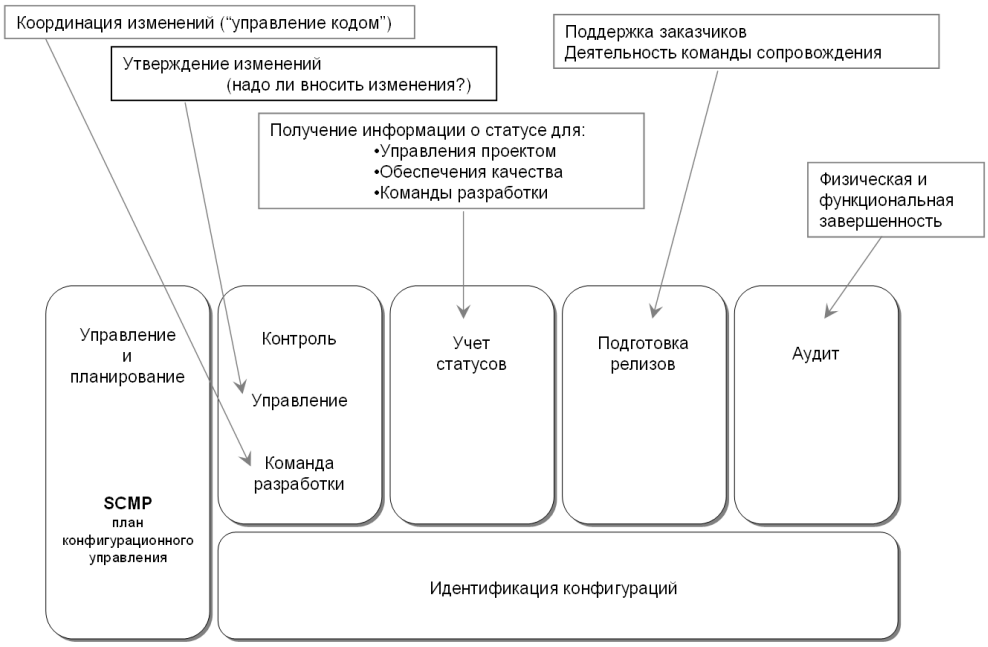

Рисунок 1. Работы по конфигурационному управлению (SCM Activities) [SWEBOK, 2004, с.7-1, рис. 1]

Данная область знаний связана со всеми другими областями знаний и дисциплинами программной инженерии, так как объектами приложения SCM являются все артефакты, создаваемые и используемые в процессах программной инженерии.

К сожалению, SCM-деятельность во многих проектных командах сводится лишь к *контролю версий* (version control) исходных текстов и, в лучшем случае, документации (причем не проектной документации, в целом, а документации на создаваемое программное обеспечение). *Попытка ограничить конфигурационное управление только вопросами контроля версий*, в какой-то степени, является результатом непонимания того, что *результаты проекта* – это не только исходный код, исполняемые модули и пользовательская документация, но и все то, что создавалось (пусть и для

решения тактических задач, как это часто бывает с некоторыми моделями и результатами пилотных работ по созданию прототипов) *на протяжении всего проекта*. Активами проекта (результатами, артефактами) являются и описания бизнес-процессов и бизнес-сущностей, и архитектурные модели, и требования, и план проекта/проектные задачи (как комплекс параметров, связанных с распределением ресурсов), и запросы на изменения (включая информацию о дефектах) и многое другое. Безусловно, упрощение вопросов конфигурационного управления до уровня управления версиями, с коньюктурной точки зрения, выгодно многим поставщикам соответствующих инструментальных средств. *В определенных случаях*, особенно, для малых проектов или временно используемых/"одноразовых" систем (например, по односторонней, "one-way" миграции данных из унаследованной системы в новую), *упрощенный взгляд на конфигурационное управление может быть вполне обоснован*. Однако, как это ни прискорбно, часто приходится наблюдать позиционирование такой, с позволения сказать, "практики", как некоего "стиля гибкой работы", подменяющей реальную динамику и гибкость agile-подходов (например, XP) отсутствием управления (важно понимать и помнить, что управление далеко не всегда является директивным), как такового (например, по определению содержания проекта на основе консенсуса проектной команды и вовлеченных в проектные работы представителей заказчика). В свою очередь, даже когда все активы проекта находятся под контролем соответствующих SCM-систем, необходимо осознавать, что *конфигурационное управление предполагает* постоянно действующий *процесс*, а не просто комплекс определенных периодически выполняемых операций. Только восприятие SCM-деятельности в качестве инфраструктурной основы процессов жизненного цикла может обеспечить эффективность управления программными проектами, то есть – достижение поставленных целей и создание результатов, удовлетворяющих заданным критериям. В то же время, *конфигурационное управление - необходимое, но не достаточное условие*, так как только совокупность процессов жизненного цикла, включая управление требованиями, проектирование и другие, не менее важные аспекты, определяют весь комплекс работ по созданию программных систем.

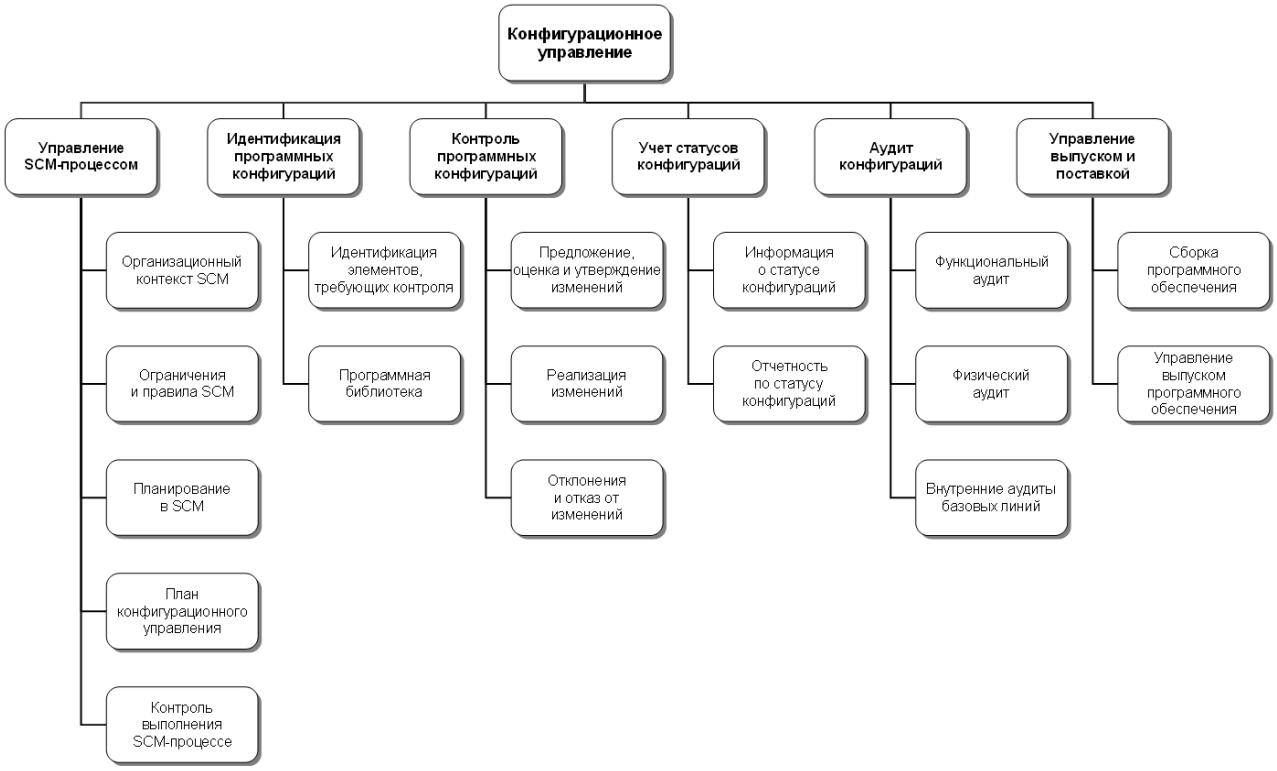

Рисунок 2. Область знаний "Конфигурационное управление" [SWEBOK, 2004, с.7-3, рис. 2]

1. Управление SCM-процессом (Management of SCM Process)

SCM-деятельность контролирует эволюцию и целостность продукта, идентифицируя его элементы, управляя и контролируя изменения, а также, проверяя, записывая и обеспечивая отчетность по конфигурационной информации. С инженерной точки зрения, SCM способствует разработке и реализации изменений. Успешное внедрение SCM требует точного планирования и управления. Это,

в свою очередь, предполагает понимание организационного контекста и тех ограничений, которые связаны с проектированием и реализацией процесса конфигурационного управления.

1.1 Организационный контекст SCM (Organizational Context for SCM)

Для планирования SCM-процесса необходимо понимать организационный контекст и связи между организационными элементами. SCM-работы предполагают взаимодействие с другими аспектами проектной деятельности (не путайте с управлением проектами – это, при всей своей значимости, лишь один из видов проектной деятельности и организационными элементами.

Организационные элементы, отвечающие за процессы поддержки программной инженерии, могут быть структурированы несколькими способами. Несмотря на то, что ответственность за выполнение определенных SCM-задач может быть назначена (принята или ассоциирована, в зависимости от управленческих принципов и установок, т.е. общего менеджмента - general management) различным частям (лицам, группам, подразделениям и т.п.) организации, например, структуре, отвечающей за разработку программного обеспечения, общая ответственность за конфигурационное управление часто возлагается на отдельный (специализированный) организационный элемент или назначенную персону.

Программное обеспечение часто разрабатывается как составная часть бо́льшей системы, содержащей аппаратные и программно-аппаратные/встраиваемые элементы. В этом случае, SCM-деятельность ведется параллельно с работами по конфигурационному управлению (CM) в отношении аппаратной или программно-аппаратной части, строго согласуясь с общим конфигурационным управлением на уровне системы, в целом. Ряд источников (см. библиографию SWEBOK, связанную с данной областью знаний) описывает SCM в сочетании с контекстом, в рамках которого проводится такая деятельность.

SCM может играть роль интерфейса к работам, направленным на обеспечение качества (quality assurance), вытекающим, например, из отслеживания записей <по изменениям> и несогласующихся элементов (например, выявленным в процессе сборки очередной версии системы, *прим. автора*). С точки зрения составителей <данной области знаний SWEBOK>, некоторые элементы, находящиеся под управлением SCM <процесса>, могут также служить объектами рассмотрения в рамках организационных программ по обеспечению качества. Управление несогласующемися элементами обычно относится к работам по управлению качеством. Однако, SCM может обеспечить существенную помощь в отслеживании (трассировке) и создании отчетности по элементам программных конфигураций, попадающих в такую <проблемную> категорию.

SWEBOK отмечает, что возможно тесное взаимодействие между организационными структурами, отвечающими за разработку и сопровождение (и SCM играет роль инфраструктуры, обеспечивающей такую связь).

В зависимости от контекста, существует множество подходов и практик в части выполнения задач конфигурационного управления в приложении к программному обеспечению. Часто, одни и те же инструменты поддерживают и разработку, и сопровождение, обеспечивая достижение целей и содержания SCM.

1.2 Ограничения и правила SCM (Constraints and Guidance for the SCM Process)

Ограничения и правила в отношении процесса конфигурационного управления порождаются различными источниками. Политики и процедуры, формулируемые на корпоративном или другом организационном уровне, могут влиять или предписывать структуру и реализацию SCM-процесса для заданного проекта. Кроме того, контракт между заказчиком и поставщиком может содержать положения, затрагивающие процесс конфигурационного управления. Например, может требоваться проведение определенных процедур проверки (аудита) или специфицирован набор элементов (активов, артефактов), передаваемых под управление <процедур и системы> конфигурационного управления (или в части формализации обработки и контроля реализации запросов на изменения, поступающих от заинтересованных лиц). Когда разрабатываемый программный продукт потенциально затрагивает аспекты публичной безопасности, могут налагаться определенные ограничения со стороны соответствующих регулирующих органов (например, USNRC Regulatory Guide 1.169, “Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear

Power Plants", U.S. Nuclear Regulatory Commission, 1997). Наконец, на структуру и реализацию SCM-процесса в проекта влияют выбранные (с точки зрения модели и адаптированных характеристик) процессы жизненного цикла и инструменты, применяемые для реализации программной системы.

Рекомендации по структуре и реализации SCM-процесса могут быть также результатом применения лучших практик (best practices), представленных в стандартах, выпущенных соответствующими стандартизирующими организациями. Лучшие практики также отражены в моделях совершенствования и оценки процессов, например, в CMMI – Capability Maturity Model Integration Института программной инженерии (SEI – Software Engineering Institute) университета Карнеги-Меллон (Carnegie-Mello University) и ISO/IEC 15504 (SPICE) "Software Engineering – Process Assessment".

1.3 Планирование в SCM (Planning for SCM)

Планирование процесса конфигурационного управления для заданного проекта должно согласовываться с организационным контекстом, соответствующими ограничениями, общепринятыми рекомендациями, а также характеристиками и природой самого проекта (например, его размером или значимостью). Основные работы, проводимые при планировании SCM-деятельности включают:

- Идентификацию программных конфигураций (Software Configuration Identification)
- Контроль конфигураций (Software Configuration Control)
- Учет статусов конфигураций (Software Configuration Status Accounting)
- Аудит конфигураций (Software Configuration Auditing)
- Управление выпуском и поставкой (Software Release Management and Delivery)

Кроме этого, необходимо принимать во внимание и такие аспекты конфигурационного управления, как организационные вопросы, обязанности, ресурсы и расписание, выбор инструментов и реализация, контроль поставщиков и субподрядчиков, а также, контроль интерфейсов <взаимодействия программных модулей>. Результаты планирования сводятся в *план конфигурационного управления* (*SCM Plan - SCMP*), обычно, являющийся объектом оценки и аудита в рамках деятельности по обеспечению качества (SQA – Software Quality Assurance).

1.3.1 Организация и обязанности (SCM organization and responsibilities)

Для предотвращения путаницы в том, кто будет выполнять заданные работы и задачи конфигурационного управления, должны быть четко идентифицированы организации (организационные структуры), вовлеченные в SCM-процесс. Конкретные обязанности по выполнению заданных работ и задач SCM должны быть назначены соответствующим организационным сущностям. Также, должны быть идентифицированы общие полномочия и пути отчетности, даже если это выполняется в процессе планирования управления проектом или деятельности по обеспечению качества.

1.3.2 Ресурсы и расписание (SCM resources and schedules)

В процессе планирования конфигурационного управления идентифицируется персонал и инструменты, привлекаемые для выполнения соответствующих работ и задач SCM. Планирование касается вопросов определения расписания, устанавливая последовательность задач конфигурационного управления и идентифицируя их связь с расписанием проекта и его вехами, определенными на стадии планирования проекта. Также должны быть специфицированы требования по обучению персонала, необходимые для реализации планов.

1.3.3 Выбор инструментов и реализация (Tool selection and implementation)

SCM-деятельность поддерживается различными типами инструментальных средства и процедур по их использованию. В зависимости от ситуации, эти инструменты могут включать комбинацию различных возможностей – автоматизированные средства могут решать отдельные задачи SCM, интегрированные средства могут обслуживать потребности многих участников процесса программной инженерии (например, SCM, разработку, проверку и аттестацию и т.п.). Значимость инструментальной поддержки конфигурационного управления (как и других аспектов деятельности в области программной инженерии) растет с каждым днем вместе со сложностью внедрения, ростом

размера проектов и сложности проектного окружения. Возможности инструментальных средств развиваются для обеспечения поддержки:

- SCM-библиотек (проектно-ориентированных баз знаний, *прим. автора*)
- Запросов на изменения (software change request - SCR) и процедур утверждения (approval)
- Управления кодом (и связанных рабочих продуктов) и изменениями
- Отчетности по статусу конфигураций и сбору соответствующих метрических показателей
- Аудиту конфигураций
- Управлению и отслеживанию <состояния и полноты> программной документации
- Выполнению задач по сборке программных продуктов и их модулей
- Управлению, контролю и поставке выпусков (релизов) программных продуктов

Инструменты, используемые для обеспечения конфигурационного управления, могут также предоставлять метрики, необходимые для совершенствования процессов. SWEBOK обращает внимание (рекомендуя соответствующий первоисточник) на следующие ключевые индикаторы: работы и прогресс <по их выполнению> (Work and Progress) и индикаторы качества – поток изменений (Change Traffic), стабильность <конфигураций> (Stability), раздробленность (Breakage), модульность (Modularity), переработка (Rework), адаптируемость (Adaptibility), среднее время между сбоями (MTBF – Mean Time Between Failures), зрелость/полнота <информации> (Maturity). Отчетность по этим индикаторам может быть организована различным образом, например, по элементам конфигураций или по типу запросов на изменения.

Рисунок 3 демонстрирует отображение инструментальных возможностей и процедур на работы по конфигурационному управлению.

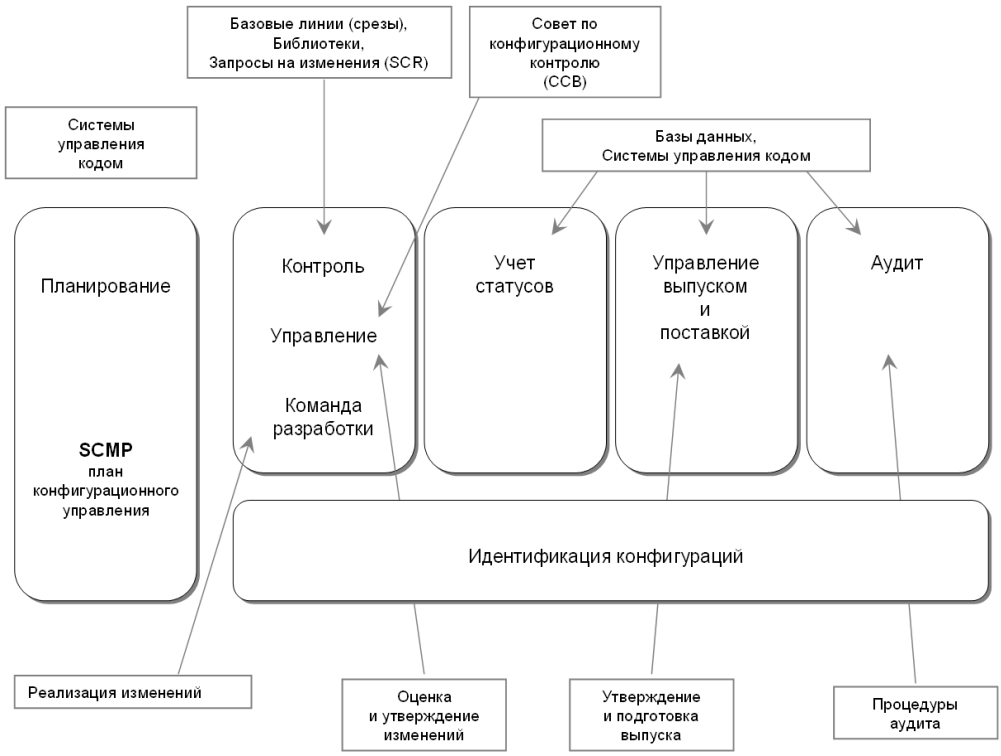

Рисунок 3. Характеристики SCM-инструментов и связанные процедуры. [SWEBOK, 2004, с.7-4, рис. 3]

В этом примере система управления кодом поддерживает программные библиотеки контролируя доступ к элементам библиотек, координирует действия множества пользователей и помогает в проведении рабочих процедур. Другие инструменты поддерживают процесс сборки и выпуска программного обеспечения и документации на основе программных элементов, содержащихся в библиотеках. Инструменты для управления запросами на изменения программного обеспечения используются для контролируемых <системой конфигурационного управления> программных элементов. Другие инструменты могут обеспечивать управление базой данных и необходимыми менеджменту отчетными средствами, а также деятельностью по разработке и обеспечению качества. Как уже упоминалось выше, в рамках SCM-системы может быть объединен целый ряд

инструментов различных типов. При этом сама система конфигурационного управления может быть тесно связана и поддерживать другие виды работ, касающиеся не только SCM.

В процессе планирования инженеры выбирают те SCM-средства, которые применимы для решения стоящих перед ними задач.

Вопрос выбора SCM-системы должен решаться исходя из целей, сформулированных в отношении используемых процессов программной инженерии и уровня зрелости этих процессов. Кроме того, необходимо учитывать и вопросы унификации программных средств, используемых для поддержки инфраструктуры разработки и сопровождения всего портфеля программных проектов, выполняемых в организации. В силу фундаментальной значимости SCM-системы для обеспечения базовых процессов программной инженерии и управления всеми проектными активами, принимать решение об использовании той или иной SCM-системы для каждого отдельно взятого проекта выглядит необоснованным. SCM-система, система управления требованиями (более чем желательно, связанная с SCM), средства бизнес-моделирования и проектирования, среды разработки – все это должно быть стандартизировано в рамках организации, за исключением тех случаев, когда требования в отношении тех или иных инструментальных средств сформулированы со стороны заказчика и являются составной частью требований, предъявляемых к проекту. Возвращаясь к вопросу выбора SCM-системы, безусловно, необходимо учитывать мнение инженеров, однако, сложившиеся привычки не должны “перевешивать” функциональность предлагаемых к унификации SCM-средств, обеспечиваемую ими доступность и прозрачность информации о состоянии проекта в любой момент времени и, конечно, возможность эффективного администрирования активов проекта, в том числе, в контексте необходимых для этого трудозатрат.

В процесс планирования рассматриваются аспекты, которые могут “всплыть” в процессе внедрения (и, даже, на этапе эксплуатации) выбираемой системы конфигурационного управления. В частности, обсуждаются и вопросы возможных “культурных” изменений, если это необходимо (с точки зрения поставленных целей – проекта и/или совершенствования процессов). Дополнительная информация, затрагивающая SCM-инструментарий, представлена в области знаний SWEBOK “Software Engineering Tools and Methods”.

1.3.4 Контроль поставщиков/подрядчиков (Vendor/Subcontractor Control)

Программные проекты могут требовать необходимости приобретать или использовать уже приобретенное программное обеспечение – компиляторы и другие инструменты (среды разработки, библиотеки компонент). Планирование должно касаться вопросов – надо и, если надо, то как помещать эти инструменты (например, интегрируя в программные библиотеки проекта) под управление SCM-системы и как их изменения и обновления будут оцениваться и управляться.

Аналогичные соображения существуют и в отношении программного обеспечения, создаваемого подрядчиками. В этом случае, в отношении SCM-процесса подрядчика предъявляются специальные требования со стороны заказчика и они вносятся в контракт, предполагая не только возможность мониторинга, но и соответствие его возможностей заданным требованиям. В последнее время все чаще отмечается важность доступности SCM-информации для эффективного мониторинга соответствия (compliance monitoring).

1.3.5 Контроль интерфейсов (Interface Control)

Когда программные элементы должны связываться с другими программными или аппаратными элементами, изменения в одних элементах могут влиять на другие элементы. Планирование SCM-процесса рассматривает, в частности, как будут идентифицироваться связанные элементы и как будут управляться и сообщаться их изменения. Конфигурационное управление может быть частью более масштабного процесса системного уровня (т.е. в рамках всей системы, к которой относятся соответствующие программные элементы) по определению и контролю интерфейсов, включая описание в соответствующих спецификациях интерфейсов, планах контроля интерфейсов и других документах. В этом случае, SCM-планирование контроля интерфейсов проводится в контексте процесса системного уровня.

1.4 План конфигурационного управления (SCM Plan)

Результаты SCM-планирования для заданного проекта определяются в *плане конфигурационного управления* (*Software Configuration Management Plan, SCMP*), который является документом, используемом в качестве описание SCM-процесса. Он всегда поддерживается в актуальном состоянии (обновляясь и утверждаясь по мере внесения в него необходимых изменений) на протяжении всего жизненного цикла. При описании SCM-плана обычно необходимо разработать ряд детальных процедур, определяющих как конкретные требования будут выполняться в повседневной деятельности.

Создание и сопровождение плана конфигурационного управления основывается на информации, получаемой в процессе работ по планированию. Рекомендации по созданию и сопровождению SCMP можно найти, например, в одном из ключевых SCM-стандартов *IEEE 828-98 "Standard for Software Configuration Management Plans"*. Этот стандарт описывает требования к информации, содержащейся в плане конфигурационного управления, а также определяет шесть категорий SCM-информации, содержащейся в плане (обычно, представленных в виде соответствующих разделов, *прим. автора*):

- Введение (Introduction) – описывает цели, содержание и используемые термины.
- Управление (SCM Management) – описывает структуру, обязанности, полномочия, политики, директивы (указания, обязательные для исполнения) и процедуры.
- Работы (SCM Activities) – определяет идентификацию конфигураций, их контроль и т.п.
- Расписание (SCM Schedule) – определяет связь работ по конфигурационному управлению с другими аспектами и процессами проектной деятельности
- Ресурсы (SCM Resources) – описывает инструменты, физические ресурсы, персонал и т.п.
- Сопровождение плана (SCMP Maintenance) – определяет правила, по которым в план вносятся изменения и описывает как эти изменения внедряются в повседневный SCM-процесс.

1.5 Контроль выполнения SCM-процесса (Surveillance of Software Configuration Management)

После того, как внедрен процесс конфигурационного управления, может быть необходимо контролировать (проводить надзор) над SCM-процессом для обеспечения того, что SCM-план исполняется надлежащим образом. В ряде случаев определяются конкретные требования по обеспечению качества (SQA), контролирующие исполнение процессов и процедур конфигурационного управления. Для этого может быть необходимо введение соответствующих полномочий и назначение обязанностей по контролю выполнения задач SCM. Аналогичные полномочия и обязанности по надзору над SCM-процессом могут существовать в контексте SQA-деятельности.

Использование интегрированных SCM-инструментов с возможностью контроля процесса может сделать процедуру надзора более легкой и прозрачной. Некоторые инструменты предоставляют высокий уровень настраиваемости для обеспечения гибкой адаптации процессов. Другие инструменты являются менее гибкими, диктуя те или иные процессы и их характеристики. Требования контроля (надзора), с одной стороны, и уровень гибкости и адпатируемости, с другой, являются определяющими критериями выбора того или иного инструмента.

1.5.1 Метрики и процесс количественной оценки в SCM (SCM measures and measurement)

Количественные показатели (метрики) могут определяться для обеспечения информации о разрабатываемом продукте или для оценки исполнения самого процесса конфигурационного управления. Связанной целью SCM-мониторинга может быть и раскрытие возможностей по совершенствованию процесса (не только SCM-процесса, но и других процессов программной инженерии). Количественная оценка SCM-процессов предоставляет хорошие средства для мониторинга эффективности деятельности по конфигурационному управлению на постоянной основе. Эти измерения полезны для оценки текущего состояния процесса и проведения сравнений во времени (как прогресса в отношении развития продукта, так и качества выполнения процесса, как такового). Анализ измерений позволяет понять причины изменения процесса и внести соответствующие коррективы в план конфигурационного управления (SCMP).

Программные библиотеки и различные возможности SCM-средств предоставляют источники для получения информации о характеристиках SCM-процесса (наравне с проектной информацией и

данными, необходимыми для принятия тех или иных управленческих решений). Например, информация о времени, необходимом для выполнения различных типов изменений, может быть полезна для оценки критериев того, какой уровень полномочий оптимален для утверждения определенных типов изменений.

Необходимо сохранять фокус на проведении анализа измерений и формировании соответствующих выводов, вместо проведения “измерений ради измерений” (к сожалению, последнее встречается слишком часто, чтобы не отметить этот факт). Обсуждение количественных оценок в отношении процесса и продукта представлено в области знаний “Процесс программной инженерии”. Программа проведения количественных оценок обсуждается в области знаний “Управление программной инженерией”.

1.5.2 Аудит в рамках SCM (In-process* audits of SCM)

Аудит может проводится на протяжении всего процесса программной инженерии для определения текущего статуса заданных элементов конфигураций или оценки реализации процесса конфигурационного управления. SCM-аудит предоставляет более формальный механизм мониторинга выбранных аспектов процесса и может координироваться с работами в области обеспечения качества (SQA; см. секцию “5. Software Configuration Auditing”).

* “in-process” подчеркивает непрерывность/периодичность аудита и его позиционирование как неотъемлемой составной части конфигурационного управления.

2. Идентификация программных конфигураций (Software Configuration Identification)

Работы по идентификации конфигураций программного обеспечения определяют контролируемые элементы (items), устанавливают схемы идентификации для элементов и их версий, а также задают инструменты и описывают техники, используемые для управления этими элементами (включая их передачу под управление SCM-процесса и системы). Данная деятельность является основой для всех других работ по конфигурационному управлению.

2.1 Идентификация элементов, требующих контроля (Identifying Items to Be Controlled)

Первый шаг в организации контроля изменений состоит в идентификации программных элементов, которые необходимо контролировать в рамках SCM-процесса. Это предполагает понимание программной конфигурации в контексте системной конфигурации, выбор элементов программной конфигурации, разработку стратегии отметки (labeling) программных элементов и описание связи между ними, а также идентификацию базовых линий (baselines, это понятие будет обсуждаться позднее в этой теме) вместе с процедурами включения элементов в базовую линию.

2.1.1 Программная конфигурация (Software configuration)

Программная конфигурация – набор функциональных и физических характеристик программного обеспечения, сформулированная в документации или воплощенная в продукте (см. IEEE 610.12-90, Standard Glossary for Software Engineering Terminology). Программная конфигурация может рассматриваться как составная часть общей системной конфигурации.

2.1.2 Элемент конфигурации (Software configuration item)

Элемент программной конфигурации (software configuration item, SCI) – фрагмент программного обеспечения, вовлеченный в процесс конфигурационного управления (и, возможно, помещенный под управление SCM-системы) и рассматриваемый как одна (атомарная) сущность в рамках SCM-процесса (см. IEEE 610.12-90). SCM контролирует множество различных элементов, включая не только программный код. Программные элементы, потенциально полезные в качестве элементов программной конфигурации (SCI), включают планы, спецификации и документы (например, полученные в результате моделирования и проектирования), программные инструменты, исходный и исполнимый код, библиотеки кода, данные и словари данных, а также документацию по установке, сопровождению, эксплуатации и использованию программного обеспечения.

Выбор SCI является важным процессом, в рамках которого необходимо достигать баланса между обеспечением адекватного уровня прозрачности представления (дословно – “видимости”, visibility) в контексте контроля проекта. Правильный выбор элементов конфигурации важен для обеспечения управляемого набора контролируемых элементов. SWEBOK дает ссылку на источник, описывающий список критериев по выбору элементов конфигураций.

2.1.3 Связи между элементами конфигурации (Software configuration item relationships)

Структурные связи между выбранными элементами конфигурации (и их составляющими) влияют на другие SCM работы и задачи, например, сборку программного обеспечения или анализ влияний (impact analysis) предлагаемых изменений. Надлежащее отслеживание этих связей является важным для поддержания актуальной трассировки (traceability) между активами проекта. Разработка схемы идентификации элементов конфигураций (SCI) должна учитывать отображение между идентифицируемыми элементами и структурой программного обеспечения, а также потребность в поддержке эволюционирования программных элементов и их связей по мере развития системы.

2.1.4 Версия программного обеспечения (Software version)

Программные элементы развиваются по мере выполнения проекта. *Версия* (*version*) программного элемента – конкретно идентифицированный и специфицированный элемент. Версия элемента может также рассматриваться в качестве определенного состояния (state) эволюционирующего элемента. *Обновление* (*revision*) – новая версия элемента, предназначенная для замены его старой версии. *Вариант* (*variant*) – новая версия элемента, добавляемая в конфигурацию без замены старой версии (то есть сосуществующая с другой версией того же элемента).

2.1.5 Базовая линия, срез (Baseline)

Базовая линия или <*фиксированный*> *срез* (*baseline*) программного обеспечения –набор элементов программной конфигурации, формально определенный и зафиксированный по времени в процессе жизненного цикла программного обеспечения. Этот термин также иногда используется для указания конкретной версии элемента конфигурации, если это согласовано заранее. В определенных случаях, базовая линия может изменяться только через формальную процедуру контроля изменений. Фиксированный срез в сочетании со всеми утвержденными изменениями в отношении его представляет собой текущую утвержденную конфигурацию.

Хотя, упоминаемая в SWEBOK классификация базовых линий в определенной степени является умозрительной и, безусловно, не единственно возможной, она полезна для адаптации и выработки внутрикорпоративных стандартов, на основе которых, в дальнейшем строятся соответствующие планы конфигурационного управления (SCMP). Приведенную ниже классификацию, соответственно, стоит рассматривать лишь как пример, но пример достаточно полезный для данного контекста обсуждения.

В общем случае, используются следующие типы базовых линий – функциональные, утвержденные, эволюционные или промежуточные, а также, базовые линии продукта. Функциональный срез соответствует принятым программным требованиям. Утвержденный срез соответствует принятым программным требованиям и требованиям в отношении интерфейсов. Базовая линия продукта представляет собой срез активов, относящихся к продукту, на заданный момент времени (при этом, базовая линия продукта не всегда является его версией, готовой к выпуску, т.е. к передаче в эксплуатацию). Полномочия по изменению заданной базовой линии обычно находятся в ведении организационной структуры, отвечающей за разработку программного обеспечения, но могут также разделяться и с другими организационными структурами (например, отвечающей за конфигурационное управление или тестирование). Базовая линия продукта соответствует завершенному программному продукту, готовому для проведения работ по интеграции в рамках целевой системы (system integration). Базовые линии, используемые для данного проекта, вместе с ассоциированным уровнем полномочий, необходимым для утверждения изменений, обычно идентифицируется в конфигурационном плане – SCMP.

Здесь уместно провести параллель между вехами (milestone) проекта и базовыми линиями. Выглядит вполне обоснованным *отображение вех проекта на базовые линии*, как “выходы”

(результаты) выполнения процессов проекта к моменту достижения соответствующей проектной вехи.

2.1.6 Включение элементов в программную конфигурацию (Acquiring software configuration items)

Различные элементы программной конфигурации передаются под управление SCM-процесса в различные моменты времени и включаются в базовые линии в определенных точках жизненного цикла. Инициирующим событием является завершение определенных форм формального утверждения задач, таких как формальная оценка (review). Рисунок 4 характеризует развитие базовой линии в процессе жизненного цикла. Этот рисунок базируется на каскадной (waterfall) модели только в целях иллюстрации; нижние индексы используются для обозначения версий эволюционирующих элементов. Запросы на изменения (software change requests, SCR), присутствующие на рисунке, описываются в теме 3.1 “Requesting, Evaluating, and Approving Software Changes”.

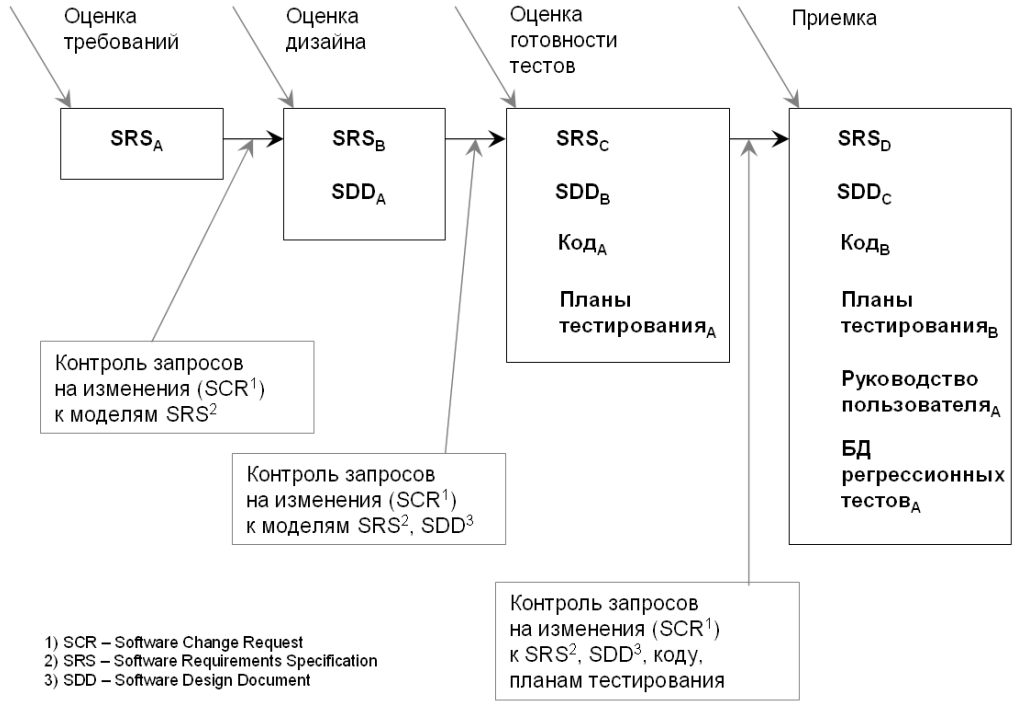

Рисунок 4. Включение элементов в конфигурацию. [SWEBOK, 2004, с.7-7, рис. 4]

После включения элемента в конфигурацию в качестве SCI, изменения элементов должны утверждаться формально, как связанные с соответствующими элементами (SCI) и базовыми линиями, следуя плану конфигурационного управления (SCMP). После утверждения <запроса на изменение и проведения работ по изменению>, <измененный> элемент включается в конфигурацию, в соответствии с заданной процедурой утверждения.

2.2 Программная библиотека (Software Library)

Программная библиотека – контролируемая коллекция программных приложений и связанной с ними документации, предназначенная для использования в процессе разработки, эксплуатации и сопровождения программного обеспечения (см. IEEE 610.12-90). В качестве <элемента> программной библиотеки, также, может рассматриваться инструментарий, используемый в работах по выпуску программного обеспечения и передаче его в эксплуатацию (например, инсталляции). На практике могут использоваться различные типы библиотек, каждая из которых соответствует определенному уровню зрелости элементов программного обеспечения. Например, “рабочая библиотека” (working library) может поддерживать работы по кодированию, “библиотека поддержки проекта” (project support library) может поддерживать тестирование, “мастер-библиотека” (master library) может использоваться для завершенных продуктов (например, как вся совокупность средств, используемых для разработки и/или выпуска продукта). С каждой библиотекой ассоциирован соответствующий уровень контроля конфигурационного управления, также ассоциированный с

базовой линией и уровнем полномочий по внесению изменений. Безопасность (в терминах контроля доступа и средств резервного копирования) является одним из ключевых аспектов управления библиотеками. SWEBOK отмечает, что существуют различные модели программных библиотек, а также приводит соответствующие первоисточники по этой теме.

Используемые для каждой библиотеки инструменты должны поддерживать контроль SCM, необходимый для данной библиотеки, как в терминах управления элементами конфигурации (SCI), так и с точки зрения контроля доступа к библиотеке. На уровне рабочей библиотеки – это средства управления кодом, обслуживающие разработчиков, специалистов по сопровождению и SCM-процесс/инструментарий (например, среда разработки должна обеспечивать интеграцию с SCM-системой). В данном контексте, рабочая библиотека фокусируется на управлении версиями программных элементов (к которым, безусловно, относится не только код, но и запросы на изменения, включая сообщения об обнаруженных дефектах, и т.п.) в многопользовательской среде. На более высоком уровне контроля, доступ ограничен сильнее и SCM (процесс и/или система) является основным пользователем <библиотеки> (например, для осуществления автоматической сборки продукта по расписанию).

Все эти библиотеки также являются важным источником информации для количественной оценки работ, их результата и прогресса <в развитии программных элементов>.

3. Контроль программных конфигураций (Software Configuration Control)

Контроль программных конфигураций касается вопросов управления изменениями в течение жизненного цикла программного обеспечения. Он включает процесс определения того, какие именно изменения должны быть сделаны, какие полномочия необходимы для утверждения определенных <типов> изменений, в чем состоит поддержка реализации этих изменений, а также концепцию формального утверждения отклонений от проектных требований, также как и отказа от внесения изменений. Получаемая в результате этого информация полезна для количественной оценки потока изменений, нарушения целостности <системы> и аспектов “переработки” в проекте (в большинстве случаев, по времени, стоимости и усилиям - трудозатратам).

3.1 Предложение, оценка и утверждение изменений (Requesting, Evaluating, and Approving Software Changes)

Первый шаг по управлению изменениями контролируемых элементов состоит в определении того, какие именно изменения надо производить. Процесс обработки запросов на изменения (software change request process), представленный на рисунке 5, включает формальные процедуры по предложению (submitting) и записи (recording) запросов на изменения, оценки потенциальной стоимости и влияния предлагаемых изменений, а также принятию, модификации или отказу от внесенных предложений по изменениям. Запросы на изменения элементов программных конфигураций могут вноситься любым лицом в любой точке жизненного цикла и может включать предлагаемые решения и соответствующий уровень приоритетности. Один из источников запросов на изменения состоит в инициировании корректирующих действий в ответ на сообщения о проблемах (problem reports). Вне зависимости от источника запроса, в самом запросе на изменение (software change request, SCR) обычно записывается информация о его типе (например, “дефект” или “заявка на расширение функциональных возможностей”/”пожелание” – enchancement/suggestion).

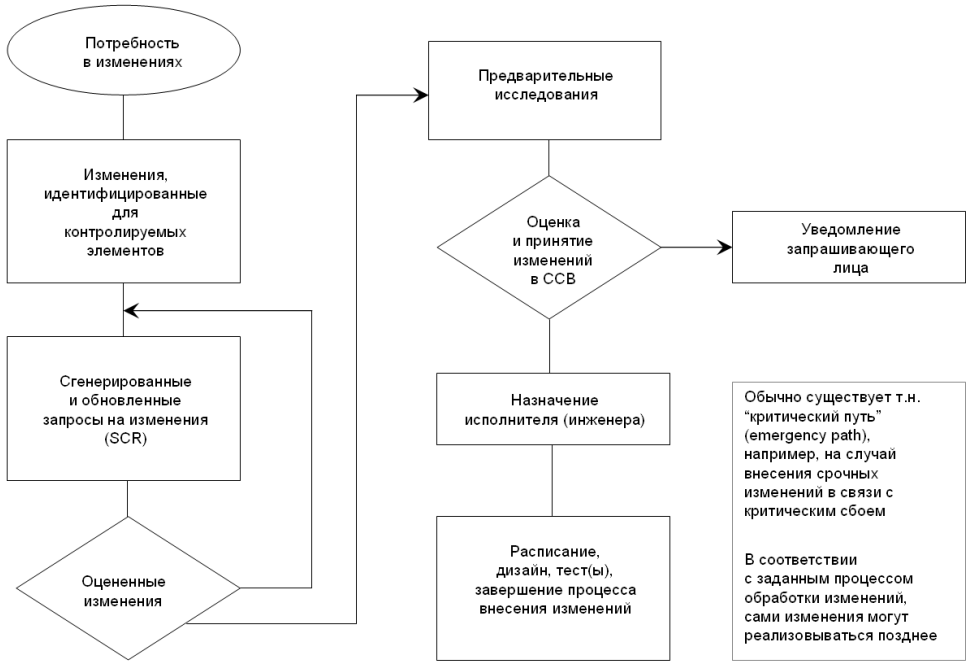

Рисунок 5. Поток процесса контроля изменений. [SWEBOK, 2004, с.7-7, рис. 5]

Такой подход обеспечивает возможность отслеживания дефектов и сбора метрических показателей о деятельности по обработке и внесению изменений, с группировкой по типу изменений. Как только получен запрос на изменение (SCR), производится техническая оценка (включающая, а иногда и называемая анализом влияний – impact analysis) запрашиваемых изменений для определения масштабов модификаций, необходимых для удовлетворения параметров запрашиваемых изменений (достаточно часто, в результате такого анализа формулируются соответствующие требования, определяющие содержание необходимых работ). Четкое понимание связей между программными (и, возможно, аппаратными) элементами системы является важной составной частью данной задачи. Наконец, органы, обладающие полномочиями, соответствующими затрагиваемой базовой линии, элементам программной конфигурации и природе изменений, должны оценить технические и управленческие (организационные) аспекты внесения запрашиваемых изменений, а также принять, модифицировать, отклонить или отложить предлагаемые изменения.

3.1.1 Совет по конфигурационному контролю (Software Configuration Control Board)

Полномочия по принятию или отклонению предлагаемых изменений обычно возлагаются на <организационную> сущность, называемую *совет по конфигурационному контролю - Configuration Control Board* (чаще звучит как совет по координации изменений - Change Control Board, CCB). В небольших проектах такие полномочия могут, в действительности, принадлежать лидеру или одному назначенному лицу из числа членов проектной команды. В общем случае может существовать несколько уровней полномочий в части принятия решений в отношении изменений, в зависимости от различных критериев, таких как критичность предлагаемых изменений, их приоритет, природа изменений (например, параметры бюджета или расписания) или текущая точка жизненного цикла. Решения о том, кто именно будет включен в ССВ для данной системы (проекта) могут варьироваться, в зависимости от данных критериев. Однако, в ССВ всегда должно присутствовать лицо, вовлеченное в организацию конфигурационного управления. В ССВ входят все заинтересованные лица, чья роль соответствует уровню ССВ. Когда содержание полномочий ССВ охватывает только аспекты программного обеспечения, такой совет называется Software Configuration (Change) Control Board – SCCB. Деятельность ССВ обычно является объектом аудита качества или оценки.

Учитывая роль управления изменениями в конфигурационном управлении и реальные задачи ССВ, более обоснованным выглядит использование именно названия *Change* Coordination & Control Board – в общем случае звучащий как *совет по координации и контролю изменений*.

3.1.2 Процесс обработки запросов на изменения (Software change request process)

Эффективный процесс <обработки> запросов на изменения (SCR process) требует использования соответствующих средств поддержки и процедур <для данного вида деятельности>, от “бумажных” форм и документированных процедур (как последовательности действий или сценариев) до программных инструментов, позволяющих отсылать запросы на изменения, выполнять процесс обработки таких запросов (изменять их статус, комментировать, детализировать и т.п.), фиксировать решения, принятые CCB, а также генерировать отчетную информацию по процессу обработки запросов на изменения. Связь между этими средствами и инструментами обработки отчетов об ошибках может существенно облегчить определение решений для обнаруженных проблем.

Обработка различных типов запросов на изменения в отношении разрабатываемых или модифицируемых программных систем, будь то сообщения о проблемах (defect report) или запросы на расширение функциональности (enchancement request), даже при разных процессах принятия решений в отношении их, должны быть объединены в единую систему (в единой базой данных), являющуюся составной и неотъемлемой частью единой среды конфигурационного управления. Только в этом случае можно обеспечить однозначную и актуальную связь между запросами различных типов и другими активами проекта (кодом, документацией, проектными моделями, задачами, ресурсами и т.п.), что позволяет не только получать актуальную информацию в любой момент времени, но и оперативно принимать те или иные (в том числе, управленческие) решения в отношении проекта, основываясь на максимально полном и корректном объеме информации.

SWEBOK отмечает, что описания процессов и соответствующих форм (например, формы сообщения о проблеме) можно найти в большом количестве внешних источников, в частности, указанных непосредственно в библиографии SWEBOK.

3.2 Реализация изменений (Implementing Software Changes)

Принятые (утвержденные) запросы на изменения (SCR) реализуются используя определенные процедуры, в соответствии с подходящими требованиями в отношении расписания. Если набор утвержденных запросов на изменения может выполняться одновременно, необходимо обеспечить отслеживание того, какие именно SCR реализованы в конкретной версии программного продукта и базовой линии <конфигурации>. Составной частью процесса “закрытия” изменения (по аналогии с closure - “закрытием”, то есть завершением проекта), является аудит(ы) конфигурации(й) и верификация качества программного обеспечения. Это обеспечивает гарантию того, что были внесены только утвержденные изменения. Описанный выше процесс обработки запросов на изменения обычно включает документирование всей информации, связанной с принятым изменением.

Фактическая реализация изменений поддерживается инструментальными средствами соответствующей программной библиотеки (см. выше 2.2 “Программная библиотека”), обеспечивающими управление версиями и поддержку <единого> репозитория кода. Как минимум, эти инструменты должны поддерживать операции chek-in/check-out (помещение в репозиторий/ получение из репозитория) и ассоциированные возможности по контролю версий (например, отметка версии – labeling, сравнение – compare/diff, слияние – merge и т.п.). Более мощные инструменты могут поддерживать параллельную разработку (parallel development) и среду географически распределенной разработки (geographically distributed environment). Эти инструменты могут объявляться (в рамках организации) как отдельные специализированные приложения, целиком находящиеся под контролем независимой SCM-группы (как самостоятельной/выделенной организационной сущности). Наконец, они могут быть столь элементарными, что включают только упрощенный контроль версий, например, на уровне файловой системы операционной среды.

Безусловно, существует широкий спектр, обоснованных функциональных возможностей инструментальных средств, используемых для решения различных задач конфигурационного управления. При этом, при выборе соответствующего инструмента или комплекса инструментов, необходимо точно понимать их функциональную нагрузку, уровень интегрируемости, возможности по адаптации с учетом конкретных процессов жизненного цикла в проектной команде или организации, в целом. Только с учетом этих критериев и других ограничений можно сформировать оптимальное и эффективное решение по программному обеспечению SCM-процесса в том объеме, который обоснован в каждом конкретном случае.

3.3 Отклонения и отказ от изменений (Deviations and Waivers)

Ограничения, накладываемые на усилия, прилагаемые к выполнению определенных работ <программной инженерии>, как и спецификации, созданные в процессе разработки, могут содержать условия, которые не могут быть удовлетворены в заданной точке жизненного цикла. *Отклонение* (*deviation*) - утверждение изменений, внесенных относительно предварительно сформулированных условий к разработке тех или иных аспектов разработки заданных элементов. *Отказ* (*waiver*) – утверждение дальнейшего использования элемента, которое отличается в той или иной степени от предварительно заданных условий. В обоих случаях используются формальные процессы (точнее, процедуры) для получения одобрения на отклонение или отказ от предопределенных ранее условий.

В большинстве случаев, говорят об *отклонении от требований* (изменении реализации требований с одновременной их корректировкой) и *отказе от выполнения запросов на изменения* (или *отклонении запросов*). Как вы уже обратили внимание, использование слова “*отклонение*” сильно зависит от контекста, подразумевая, в первом случае, определенную корректировку условий и работ и, во втором случае, полный отказ от внесения изменений с утверждением и обоснованием такого отказа.

4. Учет статусов конфигураций (Software Configuration Status Accounting)

Учет статусов программных конфигураций (Software Configuration Status Accounting, SCSA) подразумевает сохранение (recording) и генерацию отчетности (reporting) для всей информации, необходимой для эффективного управления конфигурациями программного обеспечения.

4.1 Информация о статусе конфигураций (Software Configuration Status Information)

Деятельность по учету статуса конфигураций (SCSA) предназначена и выполняется для получения (и генерации отчетов) информации, необходимой для осуществления процессов жизненного цикла системы. Как и в любой информационной системе, информация о статусе конфигураций должна идентифицироваться, собираться и поддерживаться <в актуальном состоянии> по мере эволюции этих конфигураций. Различная информация и количественные показатели необходимы для поддержки процесса конфигурационного управления, а также для генерации отчетности (о статусе конфигураций), необходимой для управления, выполнения процессов программной инженерии и других связанных видов деятельности. Типы доступной информации обычно включают идентификацию утвержденных конфигураций, наравне с идентификацией и текущим статусом реализации изменений, отклонений и отказов от изменений. SWEBOK дает ссылки на источники, содержащие возможные (частные) списки важных информационных элементов.

Современные инструментальные средства SCM должны включать определенные формы поддержки сбора и данных и подготовки SCSA-отчетности. Это может быть реализовано на уровне обращения к соответствующим базам данных, может быть представлено и в виде самостоятельных приложений, а также являться функциональной составляющей более крупных интегрированных инструментальных средств.

Логично, что только такие интегрированные многофункциональные средства возможно считать полноценными SCM-инструментами, образующими категорию *систем конфигурационного управления*. В противном случае, мы говорим лишь об отдельно взятых (пусть и взаимодействующих, в той или иной степени) инструментах - “системе управления заявками на изменения” (change request submission), “системе сообщения и отслеживания дефектов” (defect-tracking), “системе контроля версий” (version control), “системе генерации отчетности” (configuration reporting) и т.п.

4.2 Отчетность по статусу конфигураций (Software Configuration Status Reporting)

Отчетная информация может быть использована различными организационными единицами или проектными группами, включая команду разработки, команду сопровождения, управляющих проектами и персоналом, обеспечивающим проверку качества. Отчетность может предоставляться в специальной форме, следуя специфическим запросам, но может генерироваться на постоянной основе (с определенной периодичностью) в соответствии с заданными шаблонами. Определенные элементы информации, получаемой в процессе деятельности по учету статуса конфигураций, может

становиться составной частью данных, необходимых и используемых в работах по обеспечению качества (как продуктов, так и процессов).

В дополнение к отчетности по текущему статусу конфигураций, информация, получаемая в процессе SCSA-деятельности, может использоваться как основа для проведения различных количественных оценок в интересах менеджмента, разработки или конфигурационного управления. Например, к такого рода данным относятся: количество запросов на изменения на один конфигурационный элемент; среднее время, необходимое для реализации запрошенных изменений <заданного типа>.

5. Аудит конфигураций (Software Configuration Auditing)

Аудит программного обеспечения – деятельность, выполняемая для независимой оценки программных продуктов и процессов на <формальное> соответствие (conformance) применимым в данном случае инструкциям, стандартам, руководящим документам, планам и процедурам (см. IEEE 1028-97 “Standard for Software Reviews”). Аудиты проводятся в <строгом> соответствии с четко определенными процессами, содержащими и определяющими различные роли аудиторов и из обязанности. Каждый аудит должен быть, в свою очередь, четко спланирован и может требовать привлечения многих специалистов для выполнения различных задач (определяемых процедурой аудита) за достаточно короткий промежуток времени. Автоматизированные средства, обеспечивающие поддержку планирования и проведения аудита, могут существенно облегчить и ускорить этот процесс. SWEBOK отмечает, что рекомендации по проведению аудита можно найти во многих источниках, в том числе, включая стандарт IEEE 1028-97 “Standard for Software Reviews”.

Деятельность по аудиту программных конфигураций определяет степень, в которой элемент <конфигурации> (SCI) удовлетворяет заданным (например, на уровне требований и/или запросов на изменения) функциональным и физическим характеристикам. Неформальный аудит такого типа может быть связан с ключевыми точками жизненного цикла (вехами проекта, в терминах управления проектами - milestones). Существует два достаточно распространенных типа формального аудита (требуемого определенными категориями контрактов, например, на создание критически-важного программного обеспечения): *функциональный аудит* конфигураций (Functional Configuration Audit, FCA) и *физический аудит* конфигураций (Physical Configuration Audit, PCA). Успешное (в терминах соответствия результатов заданным условиям) завершение этих аудитов может быть обязательным требованиям для фиксирования базовой линии продукта. В то же время, если сравнивать контекст FCA и PCA для программного и аппаратного обеспечения, перед их выполнением необходимо четко оценивать реальные потребности в таких видах аудита (так как они требуют существенных, иногда, просто “неподъемных” затратах ресурсов, если оценивать их в рамках заданных ограничений проекта).

5.1 Функциональный аудит программных конфигураций (Software Functional Configuration Audit)

Цель FCA состоит в том, чтобы убедиться, что контролируемый программный элемент полностью соответствует заданным спецификациям. “Выход”, то есть результат проверки и аттестации (V&V, verification and validation) программного обеспечения является ключевым “входом” (исходными данными) для проведения этого аудита.

5.2 Физический аудит программных конфигураций (Software Physical Configuration Audit)

Цель PCA состоит в том, чтобы убедиться, что дизайн и документация точно согласуются с самим программным продуктом.

5.3 Внутренние аудиты базовых линий (In-process Audits of Software Baseline)

Как уже упоминалось выше, аудиты могут выполняться на протяжении всего процесса разработки для получения текущего статуса заданных элементов конфигураций. В данном случае, аудит может проводиться в отношении к выборочным элементам базовых линий с тем, чтобы убедиться, что соблюдаются заданные спецификациями характеристики процесса, скорости и качества развития продукта, а также того, что документация соответствует и поддерживается в согласованном состоянии с документируемыми элементами продукта в процессе их эволюционирования/на протяжении жизненного цикла.

6. Управление выпуском и поставкой (Software Release Management and Delivery)

Термин “*релиз*” (*release, выпуск*) используется в данном контексте, подразумевая распространение <и использование> элементов конфигураций за рамками работ по разработке программного обеспечения. Это может включать как внутренние релизы, так и выпуск и передачу программного обеспечения заказчикам. В ситуациях, когда доступны для поставки различные версии программных элементов (в частности, различные версии для разных платформ или редакции с различным набором функциональных возможностей), часто бывает необходимо создавать специализированные версии и пакеты (сборки) соответствующих материалов (элементов, активов) для выпуска в качестве <самостоятельной> версии. Программная библиотека (предоставляющая соответствующий инструментарий для такой сборки) играет ключевую роль в выполнении таких работ.

6.1 Сборка программного обеспечения (Software Building)

Сборка (building) программного обеспечения – деятельность по комбинированию корректных версий элементов программных конфигураций, проводимая с использованием соответствующих конфигурационных данных, с целью получения исполняемой программы (программной системы) для передачи заказчику и/или другим получателям (например, выполняющим работы по тестированию). Исполняемая программа для аппаратных и программно-аппаратных систем получается в результате деятельности по системной сборке (system building). *Инструкции по сборке* предоставляют описание необходимых для сборки шагов, представленных в заданной (корректной) последовательности. Работы по сборке программного обеспечения выполняются не только для получения нового релиза, но и для повторного создания предыдущих релизов в целях их восстановления, тестирования, сопровождения или каких-либо других необходимых действий.

Программное обеспечение собирается с использованием заданных версий таких средств поддержки (supporting tools), как компиляторы. В ряде случаев может требоваться повторная сборка точной копии ранее собранного элемента конфигурации. В этом случае, средства поддержки и ассоциированные инструкции по сборке должны находиться под контролем SCM (подразумевая, в зависимости от контекста, в определенных ситуациях, SCM-систему или только процесс) для обеспечения доступности корректной версии инструментария.

Часто бывают полезны возможности инструментов, позволяющие выбирать корректные для заданного окружения версии программных элементов, а также обеспечивать автоматизацию процесса сборки (например, по расписанию) программного обеспечения на основы выбранных версий и соответствующих конфигурационных данных. Такие возможности инструментов особенно необходимы для крупных проектов с параллельной разработкой и/или распределенной средой разработки (географически распределенной команды разработки). Большинство программных средств, обеспечивающих инфраструктуру разработки поддерживают такую возможность (или, как минимум, декларируют ее, прим. автора). Эти инструменты сильно отличаются (по степени комплексности предоставляемого функционала и) по своей сложности, требуя в ряде случаев изучения специализированного (специфичного для конкретного инструмента) языка сценариев, или предоставляя графические возможности, скрывающие сложность настройки “интеллектуальных” средств сборки программного обеспечения.

Процесс и результаты сборки могут быть необходимы для последующего использования <в других процессах, работах и проектах> и часто являются объектом верификации (проверки) в рамках деятельности по обеспечению качества (SQA).

6.2 Управление выпуском программного обеспечения (Software Release Management)

Управление выпуском (release management) программного обеспечения охватывает идентификацию, упаковку (сборку) и передачу элементов продукта, например, исполняемых программ, документации, аннотацию релиза (release note) и конфигурационные данные. Понимая, что изменения в продукте происходят постоянно, одной из задач управления выпуском продукта является определение момента времени, когда именно выпускать продукт (в этом контексте, управление выпуском может быть тесно связано как с деятельностью по обеспечению качества, так и с маркетинговыми соображениями в отношении выпускаемого продукта). На это решение также влияет серьезность проблем, решению которых адресуется релиз, и количественная оценка плотности сбоев (fault densities) в предыдущих релизах. Задача упаковки (packaging) состоит в идентификации того, какие

элементы продукта должны быть выпущены (например, на основании функциональных требований и их трассировки на элементы конфигурации), и в последующем выборе корректных вариантов этих элементов, задаваемом аспектами применения продукта. Документирование информации о физическом содержании релиза, обычно, включают в документ описания версии (version description document). В свою очередь, аннотация релиза (release note) содержит информацию о новых возможностях, известных проблемах, а также требованиях к платформе(ам), которые необходимо соблюдать для предусмотренного режима эксплуатации продукта. Подготовленный к выпуску пакет (package) также включает инструкции по установке и обновлению <предыдущей версии>. Создание такой инструкции может быть осложнено тем, что некоторые текущие пользователи могут иметь устаревшие версии, более ранние, чем предыдущий выпущенный релиз. Наконец, в ряде случаев, деятельность по управлению выпуском может требовать отслеживание распространения (поставки) продукта различным заказчикам или в рамках заданных целевых систем. Например, возможны ситуации, когда поставщику требуется уведомить заказчика об обнаруженных проблемах.

Для поддержки таких функций управления выпуском могут требоваться соответствующие возможности инструментария поддержки (средств поддержки или “вспомогательных средств”, если, например, попытаться максимально приблизиться к русскоязычной терминологии ГОСТ 12207). Также полезна и связь с инструментальными возможностями поддержки процесса обработки запросов на изменения для отображения содержимого релиза на полученные запросы на изменения (SCR – software change request, включая сообщения об обнаруженных ранее и исправленных в данном релизе ошибках). Эти инструментальные средства <поддержки управления выпуском продуктов> также могут поддерживать информацию о различных целевых платформах и <операционном> окружении, используемом у заказчиков.