

О терминологии и требованиях международного стандарта качества разработки программного обеспечения SW CMM

В.И.Кияев

kiyaev@tercom.ru

Санкт-Петербургский государственный университет
198504, Университетский пр., 28
Санкт-Петербург, Россия

Аннотация

Применение международного стандарта СММ позволяет поставить разработку ПО на промышленную основу, повысить управляемость ключевых процессов и производственную культуру в целом, гарантировать качество реализации ПО и исполнение проектов точно в срок. СММ становится все более востребованным стандартом среди разработчиков программного обеспечения. В связи с тем, что СММ активно используется в практике российских компаний, в настоящей статье делается попытка систематизировать ключевые понятия и требования стандарта и дать краткое толкование каждого из них. Приводятся рекомендации по формированию методов управления качеством разработки ПО в российской компании.

Введение

Для организации работ по совершенствованию методов разработки программного обеспечения с начала 1990-х гг. в США, а затем, практически, и во всем мире используется методология SW CMM [9]. Аббревиатура расшифровывается как “Capability Maturity Model for Software” или в русском смысловом варианте — “Модель зрелости возможностей [компании], разрабатывающей и сопровождающей программное обеспечение”. Эту

методологию развивает, распространяет и поддерживает Институт разработки программного обеспечения (Software Engineering Institute — SEI), входящий в состав американского университета Карнеги-Мелона.

Методология СММ разрабатывалась и развивалась в США как средство, позволяющее выбирать наилучших производителей ПО для выполнения госзаказов. Для этого предполагалось создать критерии оценки зрелости ключевых процессов компании-разработчика и определить набор действий, необходимых для их дальнейшего совершенствования. В итоге методология оказалась чрезвычайно полезной для большинства компаний, стремящихся качественно улучшить существующие процессы проектирования, разработки, тестирования программных средств и свести управление ими к понятным процедурам, отвечающим требованиям стандарта.

СММ де-факто стал именно таким стандартом. Его применение позволяет поставить разработку ПО на промышленную основу, повысить управляемость ключевых процессов и производственную культуру в целом, гарантировать качественную работу и исполнение проектов точно в срок.

Основой для создания СММ стало базовое положение, что фундаментальная проблема “кризиса” процесса разработки качественного ПО заключается не в отсутствии новых методов и средств разработки, а в неспособности компании организовать технологические процессы и управлять ими. Для оценки степени готовности предприятия разрабатывать качественный программный продукт СММ вводит ключевое понятие *зрелость* организации (Maturity) [3]. Незрелой считается организация, в которой:

- отсутствует долговременное и проектное планирование;
- процесс разработки программного обеспечения и его ключевые составляющие не идентифицированы, реализация процесса зависит от текущих условий, конкретных менеджеров и исполнителей;
- методы и процедуры не стандартизированы и не документированы;
- результат не предопределен реальными критериями, вы-

текающими из запланированных показателей, применения стандартных технологий и разработанных метрик;

- процесс выработки решения происходит стихийно, на грани искусства.

В этом случае велика вероятность появления неожиданных проблем, превышения бюджета или невыполнения сроков сдачи проекта. В такой компании, как правило, менеджеры и разработчики не управляют процессами — они вынуждены заниматься разрешением текущих и спонтанно возникающих проблем. Отметим, что на данном этапе развития находится большинство российских компаний.

Основные признаки зрелой организации:

- в компании имеются четко определенные и документированные процедуры управления требованиями, планирования проектной деятельности, управления конфигурацией, создания и тестирования программных продуктов, отработанные механизмы управления проектами;
- эти процедуры постоянно уточняются и совершенствуются;
- оценки времени, сложности и стоимости работ основываются на накопленном опыте, разработанных метриках и количественных показателях, что делает их достаточно точными;
- актуализированы внешние и созданы внутренние стандарты на ключевые процессы и процедуры;
- существуют обязательные для всех правила оформления методологической программной и пользовательской документации;
- технологии незначительно меняются от проекта к проекту на основании стабильных и проверенных подходов и методик;
- максимально используются наработанные в предыдущих проектах организационный и производственный опыт, программные модули, библиотеки программных средств;

- активно апробируются и внедряются новые технологии, производится оценка их эффективности.

СММ определяет пять уровней технологической зрелости компании, по которым заказчики могут оценивать потенциальных претендентов на получение контракта, а разработчики могут совершенствовать процессы создания ПО. Каждый из уровней, кроме первого, состоит из нескольких ключевых областей процесса (Key Process Area), содержащих цели (Goals), обязательства по выполнению (Commitment to Perform), осуществимость выполнения (Ability to Perform), выполняемые действия (Activity Performed), их измерение и анализ (Measurement and Analysis) и проверку внедрения (Verifying Implementation). Таким образом, СММ фактически является комплексом требований к ключевым параметрам эффективного процесса и способам его постоянного улучшения. Выполнение этих требований в конечном счете увеличивает возможности предприятия в достижении поставленных целей.

В результате аудита и аттестации компании присваивается определенный уровень, который при последующих аудитах может повышаться или понижаться. Следует отметить, что каждый следующий уровень в обязательном порядке включает в себя все ключевые характеристики предыдущих. В связи с этим сертификация компании по одному из уровней предполагает безусловное выполнение всех требований более низких уровней.

Приведем краткую характеристику каждого уровня.

Начальный уровень (Initial Level — Level 1)

К данному уровню относится компания, которой удалось получить заказ, разработать и передать заказчику программный продукт. Стабильность разработок отсутствует. Лишь некоторые процессы определены, результат всецело зависит от усилий отдельных сотрудников. Успех одного проекта не гарантирует успешности следующего. К этой категории можно отнести любую компанию, которая хоть как-то исполняет взятые на себя обязательства.

Повторяемый уровень (Repeatable Level — Level 2)

Этому уровню соответствуют предприятия, обладающие определенными технологиями управления и разработки.

Управление требованиями и планирование в большинстве случаев основываются на разработанной документированной политике и имеющемся опыте. Установлены и введены в повседневную практику базовые показатели для оценки параметров проекта. Менеджеры отслеживают выполнение работ и контролируют временные и производственные затраты. В компании разработаны некоторые внутренние стандарты и организованы специальные группы проверки качества (QA). Изменения версий конечного программного продукта и созданных промежуточных программных средств отслеживаются в системе управления конфигурацией. Имеется необходимая дисциплина соблюдения установленных правил. Эффективные методики и процессы институционализируются (устанавливаются), что обеспечивает возможность повторения успеха предыдущих проектов в той же прикладной области.

Определенный уровень (Defined Level — Level 3)

Уровень характеризуется детализированным методологическим подходом к управлению (т.е. описаны и закреплены в документированной политике типичные действия, необходимые для многократного повторения: роли и ответственность участников, стандартные процедуры и операции, порядок действий, количественные показатели и метрики процессов, форматы документов и пр.). Для создания и поддержания методологий в актуальном состоянии в организации уже подготовлена и постоянно функционирует специальная группа. Компания регулярно проводит специальные тренинги для повышения профессионального уровня своих сотрудников.

Начиная с этого уровня, организация практически перестает зависеть от личностных качеств конкретных разработчиков и не имеет тенденции опускаться на нижестоящие уровни. Такая независимость обусловлена продуманным механизмом постановки задач, планирования мероприятий, выполнения операций и контроля исполнения. Управленческие и инженерные процессы документированы, стандартизованы и интегрированы в унифицированную для всей организации технологию создания ПО. Каждый проект ис-

пользует утвержденную версию этой технологии, адаптированную к особенностям текущего проекта.

Управляемый уровень (Managed Level — Level 4)

Уровень, при котором разработаны и закреплены в соответствующих нормативных документах количественные показатели качества. Более совершенное управление проектами достигается за счет уменьшения отклонений различных показателей проекта от запланированных. При этом систематические изменения в производительности процесса (тенденции, тренды) можно выделить из случайных вариаций (шума) на основании статистической обработки результатов измерений по процессам, особенно в хорошо освоенных и достаточно формализованных процессных областях.

Оптимизирующий уровень (Optimizing Level — Level 5).

Для этого уровня мероприятия по совершенствованию рассчитаны не только на существующие процессы, но и на внедрение, использование новых технологий и оценку их эффективности. Основной задачей всей организации на данном уровне является постоянное совершенствование существующих процессов, которое в идеале направлено на предотвращение известных ошибок или дефектов и предупреждение возможных. Применяется механизм повторного использования компонентов от проекта к проекту (шаблоны отчетов, форматы требований, процедуры и стандартные операции, библиотеки модулей программных средств).

Отметим, что ключевые области процесса (КОП) сами не являются процессами. КОП — это статичная структура, описывающая внутренние атрибуты процесса (такие, как, например, наборы целей, обязательств, ключевых практик для каждой КОП). Ключевые области не определяют, из каких именно процессов и стандартных процедур должна состоять та или иная ключевая практика (Key Practice). Однако для того, чтобы процесс считался полностью реализованным, необходимо выполнить все требования каждой КОП.

Процессы, составляющие единый процесс разработки ПО и наполняющие содержанием ключевые области, напротив, динамичны, они развиваются и становятся более зрелыми от уровня к

уровню. Это видно из таблицы, где КОП распределены по уровням зрелости и по категориям процесса разработки ПО (Process Categories) и где можно проследить эту динамику.

Категория “Управленческая (Management)” включает действия по управлению проектом, которые эволюционируют от управления планированием и отслеживания изменений на уровне 2 к интегрированному управлению процессом на уровне 3, к управлению с использованием количественных показателей на уровне 4 и к управлению в постоянно изменяющейся окружающей среде на уровне 5.

Категория “Организационная (Organizational)” содержит взаимосвязанные атрибуты с точки зрения организационной зрелости — от фокуса на организационных проблемах единого процесса на уровне 3 к пониманию этих проблем на базе количественных показателей на уровне 4 и к достижению кульминации в значимости организационных изменений в постоянно совершенствующемся процессе и его окружении на уровне 5.

Категория “Техническая (Engineering)” содержит технические аспекты и процедуры, такие, как анализ требований, специфицирование, проектирование, кодирование, тестирование, которые выполняются на всех уровнях, но эволюционируют от чисто технических дисциплин на уровне 3 к количественному анализу и контролю программного продукта на уровне 4 и к постоянно измеряемому улучшению процессов и технологий на уровне 5.

Международный стандарт SW CMM активно используется российскими разработчиками программного обеспечения, однако до сих пор нет единого подхода к использованию ключевых терминов SW CMM. В настоящей статье делается попытка систематизировать ключевые понятия стандарта и его основные требования по каждому из уровней и дать их краткое толкование. При этом автор обобщает опыт многочисленных дискуссий на конференциях, тренингах и Internet-форумах, посвященных применению CMM в России.

Распределение ключевых областей по уровням зрелости в соответствии с категориями процесса

Уровень зрелости	Категория процесса разработки ПО		
	Управленческая	Организационная	Техническая
Начальный	Неорганизованные и случайные процессы ("как есть")		
Повторяемый	Управление требованиями. Планирование проекта разработки ПО. Отслеживание и контроль проекта разработки ПО. Управление субподрядом разработки ПО. Обеспечение качества разработки ПО. Управление конфигурацией продукта.		
Определенный	Интегрированное управление разработкой ПО. Межгрупповая координация.	Настройка процесса организации. Определение процесса организации. Программа обучения.	Технология разработки ПО. Экспертные (совместные) оценки.
Управляемый	Количественное управление процессом		Управление качеством разработки ПО
Оптимизирующий	Управление изменением процесса		Предотвращение дефектов
		Управление изменением технологий	

1 SW CMM: терминология и основные понятия

В онтологическую основу CMM положен ряд фундаментальных понятий [4, 5], которые должны одинаково интерпретироваться и применяться всеми участниками процесса разработки программного обеспечения, вовлеченными в проект. Эти понятия составляют базовую структуру понятий, используемую при определении сути ключевых областей процессов и уровней зрелости организации.

Уровень зрелости (Maturity Level). Это четко определенная ступень развития зрелости процессов разработки ПО и предприятия в целом. CMM представляет модель, состоящую из пяти уровней зрелости.

Установление (Institutionalization). Полное внедрение разработанного процесса или технологии до их становления частью организационной, функциональной и производственной сути организации и ее корпоративной культуры.

Зрелость процесса (Process Maturity). Зрелость процесса характеризует степень определенности, организруемости, наблюдаемости, управляемости, контролируемости и эффективности процесса. Понятие “зрелость” — индикатор полноты процесса или технологии, а также степени последовательности (настойчивости) организации в ее применении на всех проектах. Зрелость определяет потенциал дальнейшего совершенствования процесса.

Возможность (потенциал) процесса (Process Capability). Возможность процесса разработки ПО характеризуется способностью процесса реализовать ожидаемый результат с приемлемым диапазоном отклонений. Возможность процесса служит единым базовым показателем по предсказанию наиболее вероятных выходов проектов, планируемых в организации для выполнения.

Производительность процесса (Process Performance). Реальные параметры, достигнутые организацией, использующей данный процесс и применяющей заданную технологию.

Это понятие ассоциируется с уже выполненными проектами, разработанными внутренними стандартами, накопленными формализованными метриками и количественными показателями.

Ключевая область процесса (Key Process Area). Каждый уровень зрелости состоит из перечня ключевых областей процесса (КОП). Каждая область определяет набор взаимосвязанных процессов и технологий, реализуемых ключевыми практиками (Key Practice), совместное применение которых обеспечивает достижение основных целей (Goals), необходимых для повышения производительности и управляемости процесса до некоторого стандартного уровня, характеризующего соответствующий уровень зрелости организации. КОП были выделены таким образом, чтобы их совокупность определяла конкретный уровень.

Цели (Goals). Формируют основные ориентиры для действия ключевых практик и могут использоваться для определения эффективности внедрения организацией или проектом требований этой КОП. Цели задают назначение, границы и объем каждой КОП. Пример цели из КПО уровня 2 “Планирование проекта (Software project planning)”: “Оценочные значения при разработке ПО документируются для использования в планировании и отслеживании проекта разработки”.

Основные атрибуты (Common Features). Содержание каждой КОП разделено по пяти секциям основных атрибутов: обязательство по выполнению (Commitment to Perform), способность к выполнению (Ability to Perform), выполняемые действия (Activities Performed), измерения и анализ (Measurement and Analysis) и проверка внедрения (Verifying Implementation). Основные атрибуты отражают требования данной КОП, реализация и внедрение которых способствуют повышению эффективности, повторяемости и устойчивости процессов.

Обязательство по выполнению (Commitment to Perform). Этот атрибут описывает действия, которые необходимо предпринять организации для обеспечения гарантии, что

созданный процесс будет идентифицирован и установлен. Обычно содержит заявление политики в области качества организации и поддержки этой политики высшим руководством организации.

Способность к выполнению (Ability to Perform). Атрибут описывает начальные условия и возможности, которые должны существовать в организации для успешного выполнения проекта разработки ПО. Обычно включает технические ресурсы, финансирование, организационную структуру и обучение.

Выполняемые действия (Activities Performed). Атрибут описывает роли и процедуры, необходимые для выполнения требований КПО. Обычно охватывает установление формальных и неформальных планов, разработку процедур, выполняемые работы, контроль изменения (управление конфигурацией) базовых линий, отслеживание и корректировку процессов и работ в случае необходимости.

Измерения и анализ (Measurement and Analysis). Атрибут закрепляет необходимость измерения параметров процесса и анализа результатов измерения. Обычно включает в себя описание метрик процессов, методику измерения и анализа, а также примеры измерений, которые необходимо провести для определения статуса и эффективности выполненных действий.

Проверка внедрения (Verifying Implementation). Атрибут описывает шаги, обеспечивающие выполнение работ в соответствии с установленным процессом. Обычно включает проверки и аудиты, проводимые руководством компании, куратором проекта и группой внутреннего аудита.

Ключевые практики (Key Practices). Каждая ключевая область процесса описывается посредством ключевых практик (КП), реализация которых помогает достичь целей этой КОП. Ключевые практики описывают инфраструктуру, необходимые действия и операции, которые во многом способствуют эффективному внедрению и установлению КОП. Например, одной из ключевых практик КОП уровня 2 “Планирование проекта” является “План проекта

по разработке ПО разрабатывается в соответствии с документированной процедурой”.

Подпрактики (Subpractices). Подпрактики (подчиненные практики) приводятся под текстом ключевых практик и описывают, что ожидается от организации по выполнению данной КП. Подпрактики могут использоваться для уточнения или более детального толкования требований ключевых практик.

Положения политики (Policy Statements). Положения [установления, утверждения] политики обычно относятся к тому, что проект должен следовать документированной политике организации, касающейся использования ключевых практик по той или иной КОП. Это подчеркивает связь между корпоративными обязательствами и выполняемыми проектами. Положения политики обычно резюмируют деятельность, которая описывается в КОП и используется для идентификации и установления процессов посредством разработанных технологий, метрик и стандартов. Положения политики обычно “заточены” на те или иные аспекты деятельности организации. Так, например, в КОП “Настройка процесса организации” (Organization Process Focus) фокус направлен на организацию процесса как смысловой единицы, а не на конкретный проект.

Лидерство (Leadership). В некоторых КОП атрибут “Обязательство по выполнению” содержит утверждение, связанное с назначением роли лидера (например руководителя проекта разработки) либо с действиями распределителя ресурсов, необходимыми для успешного установления КОП. В ходе работы по проекту может появиться неформальный лидер, однако “назначенный” лидер несет всю полноту ответственности за выполнение порученной ему части проекта.

Ресурсы и финансирование (Resources and Funding). Большинство КОП содержит ключевые практики, отражающие необходимость в материальных и иных ресурсах и финансировании работ в соответствии с требованиями данной КОП. Эти ресурсы и финансирование, описанные в подпрактиках, обычно делят на три категории: доступ к специаль-

ным знаниям и документам, адекватное финансирование работ и обеспечение исполнителей требуемыми инструментами и средствами. Различные инструменты, используемые при выполнении работ, указываются в описании методик и процедур на конкретных примерах.

Обучение (Training). Обучение проводится для того, чтобы улучшить навыки сотрудников при помощи специализированных практик, моделирующих различные ситуации в ходе выполнения процесса. Такое обучение может проводиться формальными и/или неформальными методами, направленными на передачу накопленных знаний и опыта менее подготовленным сотрудникам организации. Предполагаются следующие виды обучения: аудиторные занятия, семинары по тематике проекта, наличие специализированных учебных пособий, видеокурсы, компьютеризированные инструкции и библиотеки стандартов, прямое и косвенное наставничество. Соответствующие ключевые практики описывают конкретные действия, связанные с реализацией различных видов обучения.

Для отражения действий по обучению СММ обычно используются два словосочетания: на уровне 2 — Receive Training, а на уровне 3 — Receive Required Training. Основное различие между этими терминами заключается в том, что в организации, соответствующей уровню 2, обучение не является установившимся процессом и оно производится только в случае необходимости для выполнения конкретных работ по проекту. На уровнях 3 и выше ключевые практики КОП “Программа обучения” управляют всеми действиями по обучению в организации независимо от содержания текущих проектов. В тексте стандарта СММ во всех КОП темы потенциального обучения выделены в рамки с примерами. Это подчеркивает, что различные организационные ситуации требуют различных подходов к обучению.

Ориентация (Orientation). В некоторых КОП присутствуют ключевые практики, содержащие этот термин. Ориентация — это обзор или введение в какую-либо область деятельности для описания выполняемых процедур или взаимодей-

ствия с ответственными исполнителями с целью контроля выполнения этой деятельности. Термин «ориентация» используется в широком смысле, для отражения меньшей глубины передаваемых знаний и опыта, чем при обучении.

Начальные (необходимые) условия (Prerequisite Items).

Некоторые ключевые области процесса включают ключевые практики, которые отражают необходимость в начальных условиях. Например, ключевые практики КОП “Планирование проекта разработки ПО” уровня 2 являются начальным условием для КОП того же уровня “Отслеживание хода проекта и контроль”. В других случаях они являются объектами, получаемыми извне по отношению к реализуемому проекту (например, системные требования в отношении ПО являются начальным условием для КОП “Управления требованиями” и т.д.). В СММ, где ключевые практики играют решающую роль, не все начальные условия перечислены по каждой КОП. Приведены только те, которые являются обязательными (базовыми) или достаточно критичными для внедрения той или иной КПО.

Дополнительная информация (Supplementary Information).

Состоит из примеров, уточнений и ссылок на другие КОП. Эта полезная информация выделена рамками и приводится по тексту ключевых практик.

2 Ключевые требования по уровням СММ

Ключевые требования стандарта распределены по всем пяти уровням, они отражают основное содержание ключевых областей процесса. Естественно, что при организации процесса разработки ПО могут быть выявлены и другие требования, связанные с особенностями предметной и процессной областей компании — СММ фиксирует минимальный обязательный набор, они считаются критичными, и удовлетворение этих требований при аудите деятельности компании неукоснительно проверяется. Требования перечислены ниже по уровням зрелости и всем ключевым областям процесса.

Level 1. Initial

1. **Требования отсутствуют** (No requirements). Практически неуправляемая разработка. Успех достигается только личными качествами персонала, некоторыми имеющимися заделами, долей везения.

Level 2. Repeatable

2. **Управление требованиями** (Requirements management). Требования к системе разрабатываются, анализируются, обсуждаются всеми заинтересованными группами и лицами. Они охватывают системные, функциональные, технические, ресурсные соглашения. Требования составляют единое целое с планами реализации разработки ПО. Если требования изменяются, то соответственно изменяются планы. Требования документированы в обязательном порядке. Анализ и управление требованиями являются составной и важнейшей частью проектирования программных систем [1].
3. **Планирование проекта разработки ПО** (Software project planning). Все задачи по реализации проекта документируются в виде обоснованных проектного и поэтапных (календарных) планов. Планирование включает в себя проведение всесторонней оценки работ, в том числе оценку рисков, описание ограничений, распределение и закрепление обязательств. В ходе выполнения работ планы оперативно уточняются и в случае необходимости пересматриваются.
4. **Отслеживание и контроль проекта разработки ПО** (Software project tracking and oversight). Состояние проекта (выполнение задач по этапам выполнения) отслеживается высшим руководством и менеджерами проекта и документируется по результатам периодических контрольных обзоров. При необходимости вносятся требуемые коррективы и пересматриваются текущие планы.
5. **Управление субподрядом разработки ПО** (Software subcontract management). Оцениваются возможности и

выбираются квалифицированные субподрядчики. Менеджеры и исполнители достигают договоренности о порядке выполнения, сроках и стоимости субподрядных работ, применяемым технологиям и стандартах. Все согласованные договоренности и рабочие планы документируются. Задача субподряда, сформулированная в виде частного технического задания, может быть принята в план проекта только при обоюдном согласии.

- 6. Обеспечение качества разработки ПО (Software quality assurance).** Группа контроля и обеспечения качества (Quality Assurance Group) участвует в работах по проекту с самого начала на стадиях анализа требований, планирования, согласования стандартов, прототипирования. Она должна постоянно сравнивать промежуточные параметры разрабатываемого продукта с установленными требованиями во время всего жизненного цикла и проводить текущий аудит процессов и процедур. Результаты тестирования и аудита документируются и сообщаются высшему руководству, менеджерам и исполнителям.
- 7. Управление конфигурацией продукта (Software configuration management).** Управление конфигурацией обеспечивает целостность разрабатываемого продукта в течение всего жизненного цикла. Для этого устанавливается перечень промежуточных программных продуктов и их состояние в заданные рабочим планом моменты времени. В обязательном порядке отслеживаются и документируются все изменения в различных версиях. Должно существовать полное описание отличий продукта от базовой версии, которая задается базовыми линиями конфигурации (Configuration Baselines).

Level 3. Defined

- 8. Настройка процесса организации (Organization Process Focus).** Настройка основного процесса включает достижение и поддержку должного уровня понимания ключевых организационных и производственных составляющих всеми участниками и координацию работ по управлению, оценке, разработке, сопровождению и улучшению этих составляю-

щих. Она проводится в соответствии со стандартизированной методологией ведения проектов, в которой учитывается организационная структура компании (создаются необходимые подразделения, рабочие группы). Ведется общий учет как проектной информации, так и организационных мероприятий — таких, как апробирование и внедрение новых методик и средств, обучение исполнителей. Производится экспертная и стоимостная оценка задач, новых средств разработки, сводятся результаты обучения, собираются планы всех групп. Информация должна быть размещена в соответствующих базах данных. На основе накопленной статистики производится сравнительный анализ возможностей организации.

9. **Определение процесса организации** (Organization Process Definition). Включает в себя создание и сопровождение стандартного процесса разработки ПО. Определяются организационные методы поддержки жизненного цикла процессов. Процессы идентифицируются, стандартизируются и документируются. Документы и статистика по организации процесса собираются в тематические библиотеки в едином хранилище, доступном всем разработчикам. Периодически по результатам статистических наблюдений за выполнением планов и затрат на работы производится пересмотр элементов стандартного процесса. В результате компания получает долгосрочные выгоды и преимущества.
10. **Программа обучения** (Training Program). Согласно потребностям проектов составляется программа обучения персонала. Ведется учет результатов тестирований и сертификаций. Производится периодический аудит и пересмотр программ обучения. В ходе подготовки к проекту должно быть зарезервировано достаточно времени на проведение необходимого обучения. Недостаточная квалификация исполнителей приведет, в конечном счете, к неуправляемости проекта и некачественному результату.
11. **Интегрированное управление разработкой ПО** (Integrated Software Management). Цель КОП — гармоничное согласование управленческих и производственных аспектов в рамках единого процесса. При планировании

проектов используется документированный стандарт процесса разработки плана всего проекта и этапов выполнения. Проектные и рабочие планы строятся на основе организационной структуры проекта. Разрабатываемые планы должны учитываться в специальной межпроектной базе данных. По документированной процедуре, прилагающейся к технологии разработки, используется оценка рисков, критического пути, стоимости и продолжительности работ. Должны быть шаблоны документов для технического задания, договоров, отчетов и т.д. Для принятия задачи к исполнению требуется утверждение всех заинтересованных сторон, включая заказчика. Проведение переговоров, обсуждение сроков и цен ведется на базе стандартных нормативов.

12. Технология разработки ПО (Software Product Engineering). Применяется разработанная, апробированная на предыдущих проектах и тщательно документированная технология разработки программного продукта. Стандартизированные технологические стадии используются при планировании работ. Методология процесса должна описывать:

- аналитику (документ требований к ПО, архитектуру системы, модели, методы декомпозиции, сценарные описания действий, прототипирование);
- систему хранения исходных текстов версии системы;
- кодирование (структурирование и повторное использование компонент);
- тестирование и верификация: модульное и интеграционное тестирование, системное тестирование, инспекции кода, тесты соответствия (перечисленные требования к тестированию являются обязательными);
- принципы и организацию документирования проектной и производственной документации.

На базе документированной технологии строятся проектный и рабочий планы. Ведется учет дефектов, выявляемых тестированием и совместными контрольными осмотрами.

ми (Peer Reviews) технологического состояния проекта различными группами разработчиков. Обязательно планируются контрольные точки выполнения проекта (Milestones for Control), в которых производится сравнение фактических параметров процессов с их контрольными значениями.

13. **Межгрупповая координация (Intergroup Coordination).** Требования заказчика предварительно изучаются и утверждаются для реализации всеми заинтересованными группами разработки. Требования закрепляются в техническом задании и принимаются одновременно с соглашением между группами о реализации работ. Аналитики, разработчики и члены группы QA выявляют системные и технологические проблемы на стыке работ групп и отслеживают их устранение. Обсуждение возникающих проблем и взаимный текущий аудит производятся на обязательных совместных экспертных проработках (Peer Reviews).
14. **Экспертные (совместные) оценки (Peer Reviews).** Производится совместная системная и технологическая инспекция состояния проекта группами разработчиков, разрабатывающих и тестирующих рабочие (промежуточные) программные продукты, с целью эффективного выявления дефектов и областей, в которые необходимо своевременно внести изменения. Состав конкретных промежуточных продуктов, которые должны пройти совместную экспертную оценку, учитывается в проектном плане разработки.

Level 4. Managed

15. **Количественное управление процессом (Quantitative Process Management).** Целью этой КОП является установление контроля над процессом на основе количественных показателей. По учетной информации, обеспечиваемой применением внутренних стандартов, ведется всестороннее статистическое обследование и сопровождение процесса. Разрабатываются стандартные метрики и вычисляются эталонные статистические количественные показатели стандартного процесса разработки. На основе сравнения текущих показателей с эталонными ведется управление проектами. Построив статистические характеристики по

различным технологическим стадиям проекта, получают реальные, проверенные практикой оценки параметров процессов. В дальнейшем их используют в качестве эталонных при планировании и архитектурном проектировании в последующих проектах.

- 16. Управление качеством разработки ПО (Software Quality Management).** Достижение целей этой КОП производится на основе разработки планов мероприятий в области качества, позволяющих реализовать поставленные цели. Проводится постоянное отслеживание и согласование планов процесса разработки с планами реализации качества. Ведется статистический анализ различных параметров разрабатываемого продукта, непосредственно и косвенно влияющих на качество. Полученные данные сравниваются с модельными или эталонными показателями. На основе результатов сравнения вносятся коррекции в технологический процесс и разрабатываются методы дополнения, реинжиниринга и улучшения общей методологии реализации процесса. На основе статистической обработки данных формируются стандартные метрические показатели процесса, которые используются в дальнейшем.

Level 5. Optimizing

- 17. Предотвращение дефектов (Defect Prevention).** На этапе формулирования требований и логического проектирования проводится анализ ситуаций, при которых возможно появление дефектов. Оцениваются и просчитываются риски возникновения таких ситуаций, разрабатывается комплекс превентивных корректирующих воздействий. Ведется статистический учет причин возникновения дефектов. Выявленные причины дефектов ранжируются по приоритетам и устраняются организационными и административными мероприятиями. Для полноценного статистического анализа дефектов требуется определить и документировать виды аналитических отчетов по выявленным и исправленным дефектам, которые должна представлять служба Quality Assurance.

18. **Управление изменением технологий** (Technology Change Management). Основное содержание — непрерывное совершенствование производственных и управленческих процессов. Производится плановое апробирование новых технологий и программно-аппаратных средств. Результаты регистрируются и документируются. Производится плановое внедрение новых технологий, которые показали эффективность и высокую производительность. Разрабатываются новые внутренние стандарты и руководства на эти технологии. Их применение должно стать повседневной практикой организации.
19. **Управление изменением процесса** (Process Change Management). На базе накапливаемой статистики по надежности параметров процессов производится непрерывное изучение возможностей усовершенствования процессов с целью повышения производительности труда, уменьшения сроков разработки продукта и затрат в изменяющейся проектной среде. Для этого используются новые технологические и программно-аппаратные средства, применяемые для реализации этой КОП, с последующим распространением на остальные процессы предприятия.

3 Формирование методологии управления качеством разработки ПО

Стандарт СММ все более активно используется в практике российских компаний, особенно тех, которые работают по аутсорсинговым проектам. Выполнение изложенных выше требований, разработка ключевых практик и реализация соответствующих ключевых областей процесса — достаточно протяженное, трудоемкое и затратное дело. Однако все это совершенно необходимо, чтобы получить итеративные, измеряемые, управляемые, обкатанные на многих проектах процессы с предсказуемым результатом.

Каждая организация формирует свою методологию управления качеством — соотносясь с поставленными целями, задачами, политикой в области качества, содержанием предметной области, конкретными процессами, наличием подготовленного

персонала. Система управления качеством часто содержит уникальные процессы и разработки, являющиеся ноу-хау компании, и поэтому компании редко публикуют конкретные материалы по организации единого процесса разработки ПО и своим системам качества. Тем не менее, общие методические рекомендации доступны, и они могут быть полезны, особенно в начале пути по организации работ.

В любой компании имеются наиболее “продвинутые”, т.е. полностью идентифицированные, наиболее развитые, хорошо документированные процессы. Стандарты серии ISO 9000 [2, 6] требуют минимального набора стандартных процессов, и поэтому сертификация компании только по одному или нескольким таким процессам невозможна. Стандарт SW CMM оценивает соответствие зрелости предприятия одному из стандартных уровней [9], стандарт ISO 15504 (Software Improvement and Capability Determination — SPICE) [10] вводит уже двухкоординатную размерность: оценивается не только зрелость компании, но и зрелость отдельных процессов. Таким образом, по этим стандартам можно сертифицировать процессную область компании в целом и/или наиболее зрелые процессы.

Так, например, компания в целом может быть сертифицирована по 2-му уровню CMM, а некоторые наиболее отработанные процессы в ней — по 3-му или даже 4-му уровню SPICE. В связи с этим нами была предложена общая методика организации, улучшения и сопровождения процессов разработки программных средств [7].

1. Первично выявлять, проектировать, идентифицировать, сопровождать процессы (т.е. создавать процессный “костяк”) и на их основе разрабатывать устойчивые процедуры и внутренние стандарты лучше всего в отделах и департаментах компании, выполняющих самостоятельные проекты, на основе требований Microsoft Solution Framework (MSF) [11].
2. Документировать идентифицированные процессы и строить первоначальную структуру системы качества предприятия можно в соответствии с ISO 12207 и ISO 9000:2000). Это позволит заложить надежную (системную) основу для совершенствования процессов, развития системы качества и сертификации по ISO 9000:2000. Реальный срок подготовки

такой сертификации составляет 1,0 – 1,5 года при относительно невысоких затратах.

3. Дальнейшее улучшение и сопровождение процессов на следующем этапе логично производить на базе стандартов SW CMM и SPICE, специально предназначенных для программной индустрии. Отметим, однако, что на предыдущем шаге следует заранее определить принципы документирования процессов и сразу привести их в соответствие с требованиями CMM и SPICE, чтобы впоследствии не переписывать документацию еще раз. К счастью, в большинстве случаев это возможно, так как ключевые области одних стандартов более или менее соответствуют друг другу [8]. При подготовке к сертификации по этим стандартам необходимо учитывать, что затраты на данной стадии будут в несколько раз выше, чем на предыдущей. Это объясняется недостаточной распространенностью CMM и высокой стоимостью независимого международного аудита. Тем не менее, в конечном счете затраты окупаются эффективной работой компании и доверием заказчиков.

Отметим также, что CMM и SPICE предоставляют самые общие шаблоны и методики, которые могут быть использованы для программных процессов любого типа и которые позволяют значительно повысить управляемость разработки и качество создаваемого ПО. Мы уверены, что предложенный нами путь является вполне осуществимым, минимизирует риски и затраты на сертификацию процессов и компании в целом и, что самое главное, позволяет достичь реальных целей в области качества и обеспечить долгосрочные выгоды компании.

Список литературы

- [1] Мацяшек Л.А. Анализ требований и проектирование систем // Разработка информационных систем с использованием UML. – М.: Вильямс, 2002.
- [2] Международный стандарт ISO 9000-1-94. Часть 3: Руководящие указания по применению ISO 9001 при разработке, поставке и обслуживании программного обеспечения. — М.: ИПК Изд-во стандартов, 1995. — 34 с.

- [3] Терехов А.А., Туньон В. Современные модели качества программного обеспечения. — BYTE (RE). — 1999. — № 12. — С. 30-35.
- [4] CMU/SEI-93-TR-024 “Capability Maturity Model for Software, Version 1.1”. — www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.html.
- [5] CMU/SEI-93-TR-025 “Key Practices of the Capability Maturity Model, Version 1.1”. — www.sei.cmu.edu/publications/documents/93.reports/93.tr.025.html.
- [6] Information Technology. Software Life Cycle Processes. — ISO/IEC 12207, 1995.
- [7] Kiyayev V.I., Terekhov A.A. Software Process Improvement in Russian Company: a Case Study // Proc. of the International Workshop “New Models of Business: Managerial Aspects and Enabling Technology”. — 2001. — P. 122-130.
- [8] Paulk M.C. How ISO 9001 Compares with the CMM // IEEE Software. — 1995. — P. 74-83.
- [9] Paulk M.C, Weber C.V., Curtis B., Chrissis M. B. e.a. The Capability Maturity Model: Guidelines for Improving the Software Process. — Boston: Addison-Wesley, 1995. — 442 p.
- [10] Software Process Improvement Capabilities and determination (SPICE). ISO/IEC TR 15504-CMM, 1998.
- [11] Wilson S.F., Maples B., Landgrave T. Analyzing Requirements and Defining Solution Architectures. — Microsoft Press, 1999. — P. 26-35.