

Л.Е. Матвеева

Использование принципов *Agile* при внедрении процесса производства программного продукта, основанного на *CMMI*

Представлен подход к совершенствованию процесса производства программного обеспечения, основанного на *CMMI*, заключающийся в эффективном использовании принципов и методов гибкой разработки. Описан опыт внедрения различных практик.

An approach to the improvement of the process of production of software based on the *CMMI* is presented which consists of the efficient use of the principles and methods of the agile development. The experience of the implementation of various practices is described.

Запропоновано підхід до удосконалення процесу розробки програмного забезпечення, що базується на *CMMI*, який полягає в ефективному використанні принципів та методів гнучкої розробки. Описано досвід впровадження різних практик.

Введение. Эффективное внедрение популярной и широко практикуемой модели *CMMI* (интегрированная модель технологической зрелости организации) [1] обеспечивает постоянное совершенствование процесса производства программного продукта, который условимся далее называть словом «Процесс» с большой буквы, имея в виду культуру производства программного обеспечения в компании. Модель *CMMI* является основой для поддержки Процесса, следование которому дает возможность выпускать программные продукты в срок, с высоким качеством и в рамках установленного бюджета.

В ходе более чем 15-летнего внедрения данной модели во многих компаниях по всему миру сформировались однако и основания для ее критики. *CMMI* считается очень громоздкой и неповоротливой моделью из-за обилия документирования. К тому же требуемое *CMMI* документирование часто становится ловушкой – построение процесса в организации происходит только «на бумаге». Поэтому стоит исследовать другие процессные методики и взять из них полезные приемы и принципы для эффективного построения Процесса, ориентированного на повышение качества продукта. Например, гибкие (*Agile*) методы [2–3] не предпола-

гают создания всеобъемлющей документации; один из основополагающих принципов *Agile* – «работающий программный продукт, а не всеобъемлющая документация». Один из полезных уроков, демонстрируемых *Agile*: используйте «живые» документы, чья цель и содержание являются непосредственным следствием проектных нужд.

Конечно, документирование проектного плана, функциональных спецификаций, дизайна, спецификаций тестов необходимо, так как:

- утверждение этих документов и их проверка не возможны, когда их нет в записанном виде;

- их утверждение и проверка ведут к фиксации достигнутых соглашений и совместно принятых решений, обмену опытом и знаниями, накоплению рациональных предложений в организации и совершенствованию процессов;

- все перечисленное ведет в свою очередь к повышению качества продукта;

- акт планирования выражается в написании проектного плана, функциональных спецификаций, дизайна, спецификаций тестов – это возможность посмотреть в будущее, предугадать и предотвратить возможные проблемы и ошибки.

Однако, задокументированные процедуры, которым никто не следует, и обеспечивающие автоматизацию процессов инструментальные средства, которыми не умеют или не хотят пользоваться, не заменят кропотливой работы по внедрению и постоянному совершенствованию

Ключевые слова: процесс производства программного продукта, гибкие методы, *CMMI*, зрелый процесс, обеспечение качества, модель процесса, настройка процесса.

нию Процесса в проектах и в организации в целом. Таким образом, один из основополагающих принципов *Agile*: «взаимодействие людей, а не прохождение процессов и использование инструментальных средств» помогает указать на еще одну ловушку, в которую часто попадают компании, решившие внедрять *СММИ*.

Постановка проблемы

В данной статье представлен подход к интеграции гибких (*Agile*) методик и традиционных, основанных на *СММИ*, моделей. Такая интеграция важна, поскольку современные системы обязаны уравнивать дисциплину и гибкость в производстве программных продуктов и сервисов.

Автор делится своим практическим опытом внедрения и постоянного усовершенствования Процесса, основанного на модели *СММИ*, с использованием гибких методов и практик [4]. При этом всесторонний анализ гибких методик не является исследовательской задачей в данной статье, с таким анализом можно ознакомиться, например, в [5].

Применение методик *Agile* в зрелом Процессе

В данной статье покажем, что ряд эффективных приемов и практик, присущих гибким (*Agile*) методам, полезны для внедрения Процесса, основанного на *СММИ*, как в организации, так и на уровне проекта. Более того, системное применение большинства этих приемов и практик является признаком зрелого Процесса в организации.

Зрелый Процесс

Зрелый Процесс – базовый для всей организации, это – документируемая единая технологическая система организации. В результате проведения формальной процедуры настройки и адаптации Процесс применяется на уровне проектов с учетом их особенностей. Настройка процессов разработки всегда должна быть основана на анализе фактов и предварительно собранных метрических данных.

Выделим основные принципы зрелости Процесса организации [6]:

- процессное мышление;

- системное мышление;
- культура качества;
- глубокие знания предметной области;
- управление, основанное на фактах и данных;
- статистическое мышление.

Наличие понятий «качество» и «удовлетворенность заказчика» как основных составляющих философии организации – одна из главных концепций зрелого Процесса, установленного в этой организации, наряду с понятием культуры производства, профессионализма и институционализации, превращения производства программного продукта и услуг в организованный, формализованный, упорядоченный процесс с определенной структурой отношений, ролями и ответственностями различных уровней, дисциплиной и правилами.

Ключевыми аспектами процессного мышления являются следующие:

- производство программного продукта представляет собой серию взаимосвязанных процессов;
- ошибки в процессах являются источником большинства проблем в проектах, поэтому руководители проектов должны концентрироваться на исправлении дефектов в процессах, а не на критике (наказании) членов проектных команд;
- руководители проектов должны обеспечивать наличие всех необходимых для производства процессов.

Системный подход к руководству производством предполагает управление взаимосвязанными процессами как единой системой, что позволяет повышать эффективность в достижении бизнес-целей организации. Существует два основных пути повысить производительность – работать больше и работать разумнее, развивая возможности производственных процессов как цельной системы, имеющие свои ограничения. Ограничения первого – очевидны. Ограничения второго обусловлены временной задержкой между инвестированием в усовершенствование Процесса и пожинанием его плодов. Внедрение изменений в Процесс не происходит легко, быстро, само по себе, поэтому менеджеры

часто жертвуют будущими преимуществами в угоду немедленным краткосрочным выгодам. Нужно настраивать и усовершенствовать свой процесс разработки постепенно и при этом измерять результаты изменений.

Производство состоит из взаимосвязанных процессов, характеризующихся параметрами, значения которых варьируются. Понимание статистических вариаций – основа управления, выстроенного на метрических фактах, и постоянного усовершенствования зрелого Процесса.

Общее владение Процессом (Process ownership)

Один из принципов зрелого Процесса – общее владение Процессом инженеров организации в ходе его определения и усовершенствования (*Process ownership*). Данный принцип – прямое следствие процессного мышления и культуры качества, внедренных в организации, и подразумевает необходимость постоянной поддержки в организации такого отношения инженеров к Процессу. Они должны быть владельцами этого Процесса с соответствующими правами и обязанностями. Каким образом? Опишем ряд практиковавшихся автором [4] и воплощающих принцип общего владения Процессом методов построения работы по его определению и усовершенствованию, основанного на *СММ*, являющимися по своей сути гибкими (*agile*) методиками.

Полезно периодическое проведение самооценки Процесса с участием всех его владельцев (т.е. проектных команд). Важно, чтобы процедура самооценки Процесса была детально подготовлена, запланирована, определены цели, приоритеты и задачи на каждую итерацию этой процедуры. Должна поддерживаться обратная связь, указывающая на эффективность таких сессий, во время которых анализируется ход производственного процесса с документированием полученных уроков (*lessons learned*) и принимаются решения по будущим коррективным действиям (например, изменения в какой-либо рабочей процедуре или процессном документе). Самооценивание Процесса, сопровождающееся открытыми дискуссиями членов проектной команды, – мощный

источник усовершенствования и культивации такого отношения к Процессу, как организации его владельцев. Процедура имеет много общего с гибкой (*agile*) практикой ретроспекций команды [5].

Требуется определить в организации механизмы, позволяющие инженерам ставить цели по усовершенствованию Процесса и внедрять их в производство (что соответствует гибким (*agile*) принципам самоорганизации в проектной команде и новаторскому подходу). Эффективна практика назначения ответственных за постоянное усовершенствование того или иного процесса в организации (например, процесса системного тестирования или управления требованиями) среди опытных инженеров из групп разработки или тестирования, ответственных и подотчетных руководству.

Принцип совместного владения Процессом может быть внедрен также посредством организации в компании Группы Инженерии Процесса (*SEPG*), которая работает на регулярной основе, изучает и предлагает процессные и технологические инновации на уровне организации. Рабочие группы учреждаются для выработки и внедрения конкретных решений по определению и усовершенствованию процессов, принятых Группой Инженерии Процесса, и состоят (также, как и Группа Инженерии Процесса) из представителей всех групп, непосредственно участвующих в производстве (тестировщиков, разработчиков, менеджмента и обеспечения качества – *QA*).

Для внедрения принципа общего владения Процессом может оказаться полезным ряд социальных практик из гибких (*agile*) методов. Известно, что проектная команда, применяющая гибкие методы, должна быть достаточно информирована и наделена соответствующими правами, чтобы рассматривать и проводить изменения, необходимость в которых возникает в течение жизненного цикла проекта. Применение данных практик приводит к таким выгодам:

- эффективная совместная работа по Процессу требует, чтобы все работали по одинаковым правилам;

- чтобы не отделяться от Процесса, членам проектной команды предлагается изменить его, сделать реальным для команды;

- тяжело обнаружить дефекты в процессах, если в ходе работы не требуется, чтобы им следовали;

- разработка программного обеспечения должна приносить инженерам удовольствие, если это не так, то это означает, что Процесс внедряется неправильно. Данный вывод особенно соответствует такому принципу зрелости Процесса, как процессное мышление.

Для применения принципа общего владения Процессом требуется также определить процедуру, согласно которой все инженеры организации могли бы вносить предложения и замечания по усовершенствованию Процесса и отслеживать состояние этих предложений. При этом важна эффективная организация хранилища процессных документов, все в организации должны иметь доступ к документации и информации по усовершенствованию Процесса и возможность заносить предложения по

усовершенствованию в соответствующую базу данных. Пример статистики зрелого Процесса [6], иллюстрирующий такую необходимость: количество предложений по улучшению процессов – в год от одного до десяти на одного инженера организации; время, за которое обрабатывается одно предложение – от пяти до 20 рабочих дней; количество предложений, принятых к внедрению – 60–80 процентов от общего количества занесенных в базу данных предложений.

На рис. 1 представлена разработанная автором и применяемая на практике процедура выявления и внедрения в производство предложений по усовершенствованию Процесса (*Process Improvement Process*) [4].

Формат данной схемы, представляющей последовательность выполняемых действий и зоны ответственности, как и схемы на рис. 2, соответствует требованиям стандарта *ISO/TS 16949: 2002*. Данный стандарт основан на *ISO 9000* и является специализированным для автомобилестроения. Однако некоторые его элементы

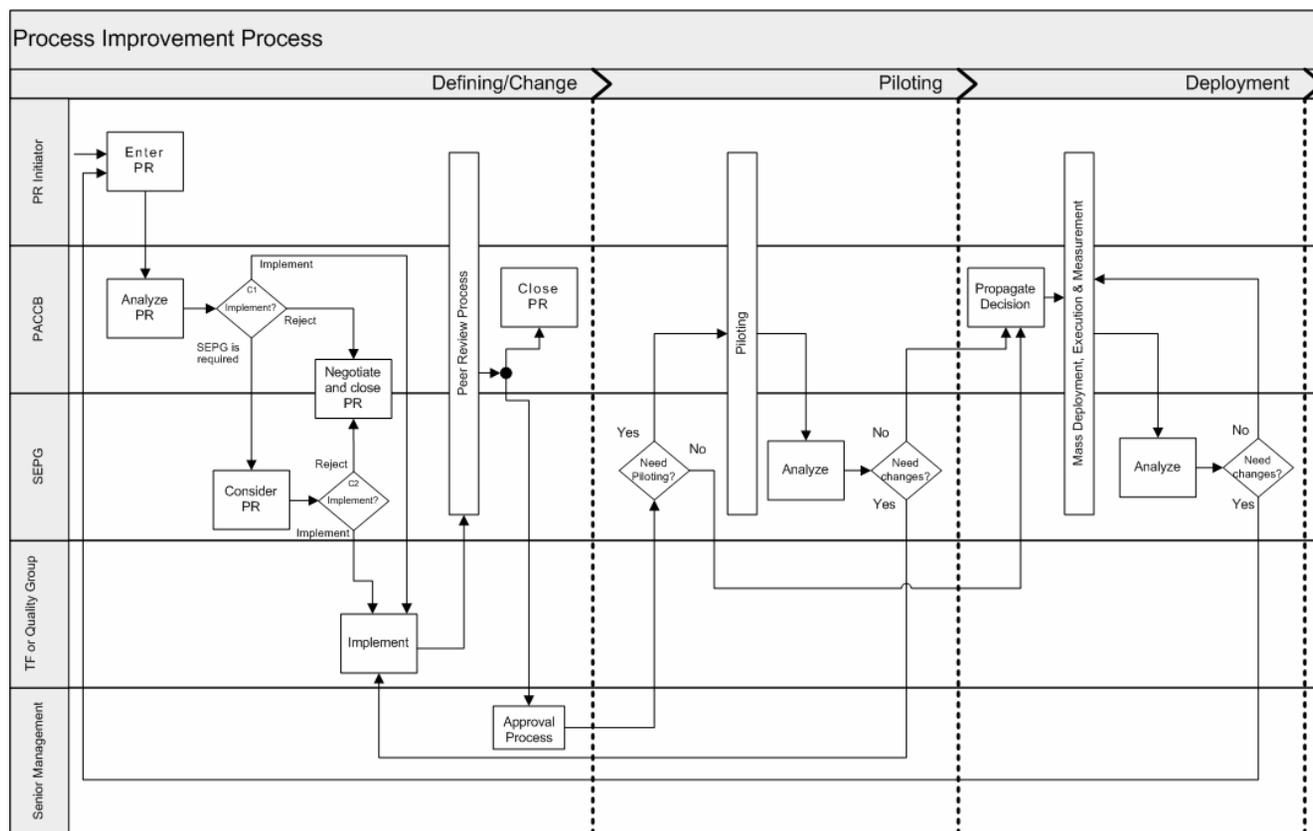


Рис. 1. Схема регистрации, утверждения и реализации предложений по усовершенствованию Процесса

(как, например, формат диаграммы активностей процесса или процедуры) могут эффективно применяться и в других областях, так как цель стандарта – развитие системы управления качеством, предусматривающей постоянное усовершенствование процессов, делая ударение на профилактике дефектов и содействуя усовершенствованию как цепочки поставки, так и процесса сертификации.

Вертикальные полосы на схеме (рис. 1) соответствуют различным фазам процедуры: определение/изменение (*Defining/Change*), пилотирование (*Piloting*) и применение (*Deployment*). Горизонтальные полосы на схеме соответствуют различным функциям (т.е. определенным зонам ответственности) процедуры реализации и применения запроса на усовершенствование процесса:

- инициатором запроса на усовершенствование процесса (*PR (Process Request) Initiator*) может быть и рядовой инженер, внесший предложение по улучшению того или иного процесса;
- Группа Инженерии Процесса (*SEPG – Software Engineering Process Group*);
- группы обеспечения качества (*Quality Group*);
- группа контроля изменений процессных артефактов – это часть группы обеспечения качества (*PACCB – process assets change control board*);

- руководство (*senior management*);
- рабочая группа (т.е. группа специалистов для решения конкретной задачи) (*TF – Task Force*); она назначается Группой Инженерии Процесса.

Условие «C1 Implement ?» (см. рис. 1) включает следующие варианты выбора:

Implement – данный вариант используется в случае, когда требуются небольшие изменения (например, вводится новый шаг в какой-либо процедуре);

SEPG is required – требуется создать нормативный документ, определяющий новую процедуру, или внести существенные изменения в имеющуюся (например, предлагается заменить инструментальные средства, поддерживающие изменяемую процедуру, или ввести в нее новую роль с новыми обязанностями);

Reject – предложение по усовершенствованию процесса отвергается после анализа по той или иной причине, которая должна быть указана.

Условие «C2 Implement ?» (рис. 1) включает следующие варианты выбора:

Implement – Группа Инженерии Процесса принимает решение по реализации предложения по усовершенствованию процесса. При этом возможно потребуются дополнительные эксперты и ресурсы для реализации того или иного предложения (тогда формируется специальная рабочая группа), в противном случае улуч-

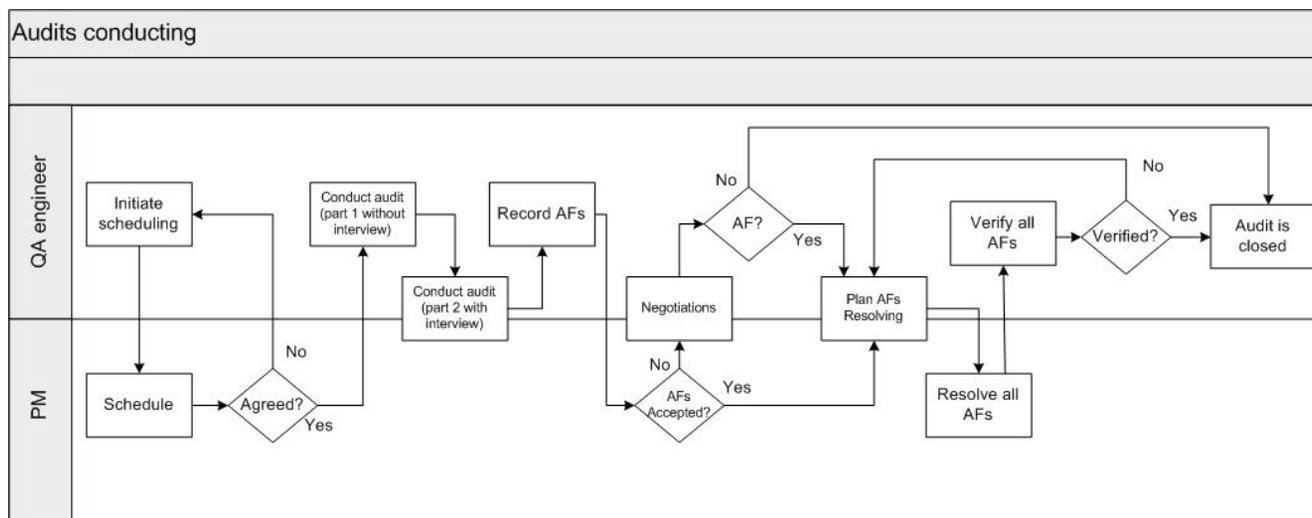


Рис. 2. Схема проведения аудита качества

шение может быть реализовано группой обеспечения качества.

Reject – предложение по улучшению отвергается после анализа по той или иной причине, которая должна быть указана.

Фазы пилотирования (*Piloting*) и применения (*Deployment*) должны включать соответствующее обучение. Пилотирование изменения того или иного процесса необходимо лишь как промежуточная фаза для анализа существенных инновационных или высокорисковых изменений.

Приведем пример внедрения принципа общего владения процессом, взятый из опыта Ст. Петербургского центра компании *Motorola* [7] по переходу с модели *CMM (level 5)* на модель *CMMI (level 5)*, который длился два года и завершился успешным оцениванием на 5-й уровень *CMMI*: более 25% сотрудников организации были вовлечены в переработку 30% и создание 10% процессных документов. Еще один пример взят автором из собственного опыта по переходу одного из подразделений компании Люксофт с модели *CMMI (level 2)* на отраслевой стандарт *ISO Automotive SPICE (level 3)*: 15% сотрудников организации вовлечены в переработку и создание 70% процессных документов. Переход был начат более года назад, плановый срок оценивания на 3-й уровень по модели *ASPICE* запланирован на 2010 год. Уровень вовлеченности сравнительно ниже, чем в предыдущем примере, что объясняется более низким уровнем зрелости Процесса.

Важным признаком зрелого Процесса является эффективная организация хранилища процессных документов. При его создании рекомендуется придерживаться гибких (*Agile*) принципов простоты и небольшого объема документации. Практика внедрения документированного Процесса показывает, что более эффективны в использовании небольшие по объему и достаточно независимые друг от друга инструкции и руководства (*guidelines*) [4, 6]. Определение каждого отдельного процесса должно быть самодостаточным и иметь минимальное количество ссылок на другие процессы. Описание всякого процесса должно занимать не бо-

лее трех страниц. Рационален такой подход к построению системы процессных артефактов, когда описания процессов производства, имеющие примерно одинаковую частоту изменения, выделяются в отдельные документы. Например, базовые правила Процесса организации, которые не должны нарушаться в результате настройки проектных процессов, и стандартный жизненный цикл проекта рекомендуется описать отдельно от таких рабочих шаблонов и инструкций, частота изменений в которых будет значительно выше [4].

Настройка и адаптация процессов (Process tailoring)

Рассмотрим такую практику зрелого Процесса, как настройка и адаптация стандартных организационных процессов в проектах (*process tailoring*).

Гибкие методики и практики могут применяться для настройки и адаптации процессов разработки программного обеспечения в отдельных проектах и в целом в организации, что обеспечивает баланс между внедрением повторяющихся процессов и возможностью применять некоторые особенности в конкретном контексте разработки. Принцип гибких методов «Реагирование на изменения, а не прохождение плана» в зрелом Процессе (который по своей природе является управляемым планом (*plan-driven*)) следует интерпретировать так, что гибкая настройка проектного процесса предпочтительнее жесткого формального следования стандартному процессу. Другой принцип гибких методик регулярного анализа проектной командой своих действий и обратных реакций на действия с целью быть более эффективными (*team reflections*) также соответствует итеративной адаптации проектных процессов, в которой принимает участие вся команда разработчиков такого проекта.

Выделим три возможные стратегии для настройки процессов:

- определение основного набора процессных документов, подходящего для разных типов разработки, для чего можно использовать всеобъемлющую, основанную на модели структуру процессов в качестве базового набора для вы-

бора соответствующих процессных элементов в начале проекта;

- определение наборов процессов, руководств и шаблонов для основных видов разработки. В рамках данной стратегии необходимо определить процедуру настройки процесса отдельного проекта и оформить ее на уровне организации в виде специального руководства с целью помочь всякой проектной команде в выборе тех процессов, которые наилучшим образом подходят для данного проекта с возможной точной подгонкой;

- определение специально адаптированного для проекта процесса путем смешивания идей и техник из лучших практик и опыта, накопленного в организации. Такая стратегия предполагает предварительное создание базового набора процессных артефактов для их выбора, основанного на особенностях проекта и продукта.

Первая и вторая стратегии являются статической настройкой процесса и характерны для традиционной разработки программного продукта, в то время как третья стратегия является динамическим подходом к настройке процессов, порожденным гибкими методиками разработки [27]. Управляемый планом Процесс традиционно воплощает всеобъемлющий подход, предполагающий необходимость адаптации под конкретную ситуацию. Настройку в этом случае могут осуществлять только эксперты, остальные предпочитают не рисковать и используют полную модель, тратя при этом лишние ресурсы. Гибкие методы предлагают другой подход: начать с относительно небольшого набора практик, расширяя этот набор итеративно только после соответствующего анализа и оценивания полученной выгоды от вложенных затрат на их внедрение.

Культура качества и обеспечение качества (QA)

Культура качества – один из главных принципов зрелого Процесса. Функции обеспечения качества (*Quality Assurance – QA*) включают:

- определение целей по качеству и плана по управлению качеством;
- отслеживание и сбор метрической информации и данных;

- проведение аудитов качества (*audits*).

Аудит качества – это объективная оценка соответствия процессов, продуктов и сервисов принятым в организации стандартам и процедурам. Объективность оценки обеспечивается использованием критериев, относительно которых идет проверка индикаторов внедрения и институционализации производственных практик. Использование критериев означает наличие списков пунктов проверки, содержащихся в контрольных листах (*checklists*) аудитов, согласованных всеми участвующими сторонами.

На рис. 2 представлена разработанная автором и внедренная в производство процедура планирования и проведения аудитов качества. Роли процедуры аудирования (горизонтальные полосы на схеме): инженер по обеспечению качества (*QA engineer*), руководитель проекта (*Project Manager* или *PM*). *AF (audit finding)* – несоответствие, выявленное в ходе аудита.

Основные шаги процедуры, представленной на рис. 2:

- Инициация привлечения аудита качества состоит в обеспечении предварительного условия его проведения: наличие списков пунктов проверки, которые содержатся в предварительно согласованных контрольных листах (*checklists*) аудитов. В свою очередь контрольный лист адаптируется к нуждам конкретного проекта и утверждается руководителем проекта. Настройка аудита и его контрольного листа проводится с целью учета специфики проекта и проведения аудиторской проверки более целенаправленно.

- Аудит планируется. План-график согласовывается.

- Аудит проводится.

- Данные аудита регистрируются. Заполняется отчет о проведении аудита после соответствующего согласования с указанием обнаруженных несоответствий и рассылается всем заинтересованным сторонам.

- Обеспечивается исправление несоответствий, обнаруженных в ходе аудита, с последующей верификацией *QA* инженером.

Деятельность по обеспечению качества требует дополнительных ресурсов и является сла-

гаемым цены качества или *COQ* (*Cost Of Quality*) [4]. Цена качества – значительная часть расходов проекта. Минимизация затрат при высоком уровне эффективности аудирования осуществляется за счет выбора оптимального подхода, во многом соответствующего гибким методам.

Функции обеспечения качества в проекте могут быть внедрены различными путями:

- активности во всех проектах организации подвергаются аудированию одновременно;
- аудит проводится только по заранее определенным видам деятельности в запланированных контрольных точках;
- аудирование проводится посредством тесного сотрудничества с командой проекта для обеспечения проверки необходимого и в правильно выбранный момент времени.

Главные недостатки первых двух подходов следующие:

- серьезные несоответствия могут быть найдены слишком поздно, их устранение будет стоить дорого;
- может быть найдено много ошибочных дефектов и несоответствий по той причине, что *QA* инженер недостаточно тесно взаимодействует с проектной командой и не имеет полной и точной информации о ходе проекта;
- аудиты не проходят настройку под проектные цели, не адаптируются к конкретным условиям, не учитывают проектную специфику. Как следствие – некоторые процессные области могут остаться вне требуемого контроля.

Третий подход предполагает, что функции обеспечения качества (*QA*) встраиваются в жизнь проекта и *QA* инженеры тесно взаимодействуют с командой. В проектной команде, практикующей гибкий метод производства программного продукта, за качество программного продукта отвечает вся команда. Такой гибкий подход позволяет избежать недостатков. Он эффективен с точки зрения снижения *COQ*. При этом обеспечивается 100%-ное покрытие проверкой всех процессных областей и унифицированный подход к аудированию в проектах. Ошибочно считать подобное вовлечение *QA* инженера уве-

личением затрат на активности по обеспечению качества. Наоборот, оно ведет к снижению *COQ*. Частота аудитов становится оптимальной. Количество ошибочно сформулированных несоответствий уменьшается, соответственно, уменьшаются временные затраты на их обработку. Превентивные действия позволяют обнаруживать проблемы как можно раньше, уменьшая затраты на их исправление.

В некоторых компаниях, внедряющих управляемый планом Процесс, встречается ситуация, когда фактически только группа обеспечения качества является владельцем Процесса и безуспешно пытается улучшить его в отрыве от остального производственного коллектива. Попавшие в такую ловушку компании отмечают отсутствие преимуществ от внедрения *СММ* [5]. В зрелом Процессе должны быть выстроены механизмы, не допускающие подобной ошибки. Необходимо постоянно практиковать в компании отношение «все являются владельцами Процесса». Кроме того, функции обеспечения качества должны внедряться в проект, начиная с фазы его инициации (происходит адаптация Процесса организации под конкретный проект). *QA* инженер не только проводит требуемые аудиты, но и вовлекается в большую часть проектных совещаний и экспертных оценок технических и управленческих документов (*peer review*), что позволяет понять специфику проекта. Глубокое знание специфики проекта позволяет *QA* инженеру выбирать оптимальное время и объем проверки.

К темам, связанным с качеством, в проектной команде, практикующей гибкие методы, обращаются во время всего жизненного цикла процесса. Статус или приоритет активностей, связанных с качеством, такой же, как и прочих активностей процесса разработки, в отличие от управляемого планом Процесса, где приоритет процессных активностей чаще всего невысокий. Стиль работы в команде, практикующей гибкие методы, – это совместная работа всех членов команды, в то время как в управляемом планом Процессе инженеры-разработчики и *QA* инженеры могут конфликтовать друг с другом в ходе выполнения своих рабочих функций.

Такой конфликт можно минимизировать, применяя гибкие практики и приемы. Во-первых, необходимо культивировать правильный подход к организации и проведению аудитов качества в проектах. Необходимо тесное взаимодействие *QA* инженера с проектной командой. Обязательным является предварительное согласование плана проведения аудита и контрольного списка, формулировок несоответствий, выявленных в ходе аудита данного проекта, и верификация требуемых коррективных действий (рис. 2). Во-вторых, специалисты по обеспечению качества должны постоянно проводить в проектных командах консультации и обучение и по процессам и процедурам. Опыт *QA* инженеров, работающих с разными заказчиками в проектах разных типов, становится очень важным на этапе настройки процесса в проекте, что позволяет наладить проектный процесс оптимальным образом и предотвратить возможные проблемы.

Рассмотрим еще один принцип гибких методов: совместная работа с заказчиком, а не согласование условий контракта в контексте процесса обеспечения качества. В качестве продукта в данном случае выступают производственные процессы, а в роли заказчика продукта – проектные команды. Согласование условий контракта – это, когда работа инженера по обеспечению качества превращается в работу полицейского, а отклонение от контракта ведет к наказанию. В действительности, проявление нарушения задекларированного процесса дает материал инженеру по обеспечению качества: случилось нарушение, почему? По большей части люди делают работу, нарушая тот или иной процесс, поскольку считают, что он несовершенный и можно сделать лучше. Следовательно, нарушение Процесса является в первую очередь источником идей по его усовершенствованию, которые инженер по обеспечению качества должен отбирать, анализировать и внедрять. Аудиты могут быть началом действий по усовершенствованию Процесса.

Таким образом, применение гибких практик способствует внедрению принципа культуры

качества – одного из главных принципов зрелого Процесса.

Процесс предотвращения дефектов (defect prevention)

К процессам, которые непосредственно отвечают за качество продукта, относятся: поиск дефектов, исправление дефектов и предотвращение дефектов (*defect prevention*). Процесс предотвращения дефектов – это анализ дефектов, идентификация корневых причин и принятие специальных мер для предотвращения повторения подобного типа дефектов. Компания, имеющая зрелый Процесс, определяет и планирует превентивные действия с целью устранения причин потенциальных несоответствий (дефектов и проблем) для предупреждения их появления [4].

Инженер по обеспечению качества обнаруживает возможности усовершенствования процессов, занимаясь поиском типов образцов низкой эффективности, дефектов или других проблем. Он ищет ответы на следующие вопросы:

- Существует ли возможность предотвращения того или иного типа (или группы) дефектов?

- Как обнаружить и исправить их более оптимальным образом?

Он изучает опыт прошлых проектов и ищет пути усовершенствования процессов разработки, при этом его интересуют ответы на такие вопросы:

- Что было хорошо? Как этот позитивный результат (*lessons learned*) повторить в будущем?

- Что было плохо? Как снизить вероятность появления подобного типа проблем или свести ее к нулю?

Проблемой традиционных дисциплинированных управляемых планом процессных моделей является недостаток механизмов, гарантирующее последующее использование уроков, полученных от завершенных проектов (*lessons learned*). Гибкие методы включают практики, например – техника ретроспекций команды или ежедневные короткие собрания *Scrum* (*Scrum meetings*), обеспечивающие принятие решений, способствующие выявлению проблем в проек-

те, а также синхронизирующие работу в команде, что обеспечивает распространение знаний от экспертов менее опытным разработчикам. Последнее обеспечивается также практиками общего владения кодом и программирования в парах (*pair programming*). Использование уроков, полученных от завершенных проектов, происходит также за счет перемещения инженеров из одного проекта в другой. Полезной может оказаться гибкая (*agile*) практика назначения в проекте «наставника» (*coach*), которого выбирают из членов проектной команды и чья функция – постоянно следить за усовершенствованием Процесса в проекте.

Дополнительные области применения гибких методов

Существующий ряд областей зрелого Процесса, где гибкие практики также могут с успехом применяться, можно дополнить:

- **Процесс планирования.** Оцениванием в проектах, практикующих гибкие методы, занимаются сами разработчики, а не менеджеры. Использование итеративной инкрементальной модели разработки с достаточно короткими итерациями (2–3 недели) дает обратную связь по поводу действительной скорости продвижения проекта и позволяет снизить риски агрессивного планирования. Для определения времени готовности к выпуску версии продукта используется процент успешно пройденных модульных и системных тестов. Все это в той или иной степени может быть применено в Процессе, основанном на *СММИ*.

- **Процесс конфигурационного менеджмента.** Гибкие (*Agile*) методы ориентированы на людей, а не процессы. Эффективность конфигурационного менеджмента повышается при смещении акцента с активностей по контролю на активности по поддержке и обслуживанию разработки программного продукта. Рекомендуется внедрять и усовершенствовать такие важные с точки зрения инженера-разработчика аспекты Конфигурационного Менеджмента, как контроль версий (*version control*), управление процессом компиляции исходного кода в исполняемый код (*build management*), управление рабочей областью (*Workspace management*), кон-

троль параллельной разработки кода (*branching management*), управление изменениями (*Change management*), управление выпуском программного продукта (*Release management*).

- **Процесс принятия решений** эффективен, если он учитывает гибкий принцип совместной работы с заказчиком. Разработка должна быть проактивной, обеспечивая заказчика альтернативными решениями в легко воспринимаемой им форме.

- **Инжиниринговые процессные области:** выработка требований и технических решений, конструирование и интеграция кода, верификация и валидация продукта – вот огромная сфера, где гибкие практики могут с успехом применяться в зрелом Процессе. Например: определение приоритетов требований в начале каждой итерации разработки; принцип простоты при разработке дизайна продукта; разработка, управляемая тестами; частый рефакторинг кода; принцип переноса кода в репозиторий только после успешного прохождения специального набора тестов; постоянная интеграция кода с регрессионным тестированием; широкое применение автоматизированного тестирования и более раннее исправление найденных дефектов; частые инкрементные поставки версий продукта.

Заключение. В данной статье представлен ряд приемов и практик, присущих гибким (*Agile*) методам, применявшихся автором при внедрении Процесса, основанного на модели *СММИ*, и продемонстрировавших свою эффективность.

СММИ можно интерпретировать двояко: либо строгий, тяжеловесный стандарт, обеспечивающий единый подход ко всем проектам в организации, либо управляемое рисками сочетание дисциплины и гибкости. Современные системы обязаны уравнивать дисциплину и гибкость в производстве программного обеспечения. Внедрение Процесса подразумевает постоянный поиск того, что эффективно работает именно в вашей компании и в конкретной ситуации.

1. *СММИ* for Development, Version 1.2. – <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>

2. *Cockburn A.* Agile Software Development. – Boston: Addison-Wesley, 2001. – 256 p.
3. *Agile Alliance* – <http://www.agilemanifesto.org>
4. *Матвеева Л.С., Волков В.А.* Процес розробки програмного забезпечення. Від теорії до практики. – К.: ИПС, 2008. – 116 с.
5. *Salo O.* Enabling Software Process Improvement in Agile Software Development Teams and Organizations. – VTT Publications 618, 2006. – 149 p.
6. *Paulk M.C., Chrissis M.B.* The 2001 High Maturity Workshop. – Software Engin. Inst., CMU/SEI-2001-SR-014. – Jan. 2002.
7. *Babkin A.V., Sevastyanov S.U., Matvienko N.I.* Moving to CMMI: Approach, Results and Lessons Learned. Proc. of St. Petersburg IEEE Chapters. – 2005. – 2. – P. 183–188.
8. *Beck K., Andres C.* Extreme Programming Explained: Embrace Change. – Boston: Addison-Wesley, 2004. – 189 p.
9. *Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун и др.* – К.: Академперіодика, 2007. – 672 с.

Поступила 10.03.2010
Тел. для справок: (044) 526-0058 (Киев)
E-mail: veeva@luxott.com
© Л.Е. Матвеева, 2010



Окончание статьи В.А. Вышинского и др.

Если требуется знание функции в узлах интервала ее задания, то вектор коэффициентов интерполяционного многочлена следует умножить на матрицу Вандермонда, соответствующую данному интервалу.

Заключение. Предложенные алгоритмы реализации процедуры интерполирования хорошо адаптированы к МА ЭВМ, т.е. они максимально используют информационные «емкости» операндов и мощности машинных команд. Таким образом, в распоряжение машинной алгебры полиномов представлены алгоритмы, «опирающиеся» не на операции алгебры действительных чисел, а на операции алгебры поли-

номов, что приводит к существенному упрощению их реализации в микроэлектронном исполнении.

1. *Вышинський В.А.* Електронні обчислювальні машини на основі алгебр з регулярним матричним представленням: Автореф. дис. д-ра техн. наук. – Київ, 2003. – 31 с.
2. *Вышинский В.А.* Об одном решении фундаментальной проблемы современного развития вычислительной техники // УСиМ. – 2003. – № 4. – С. 81–91

Поступила 22.02.2010
Тел. для справок: (044) 526-3598 (Киев)
© В.А. Вышинский, А.Ю. Кононенко, А.В. Слепец,
А.В. Вышинская, 2010



Внимание !
Оформление подписки для желающих
опубликовать статьи в нашем журнале обязательно.
В розничную продажу журнал не поступает.
Подписной индекс 71008