

Юрченко Алексей Васильевич

Yurchenko Alexey Vasyliovich

ЗАО «ЕС-лизинг»

ЕС-leasing

**Руководитель направления конфигурационного управления
и сопровождения программного обеспечения**

**Head of software configuration management
and software maintenance department**

E-Mail: E-Mail: yurchenkoalex@inbox.ru

Конфигурационное управление программного обеспечения: метрики и модель оценки окупаемости инвестиций

Software configuration management: metrics and return of investments model

Аннотация: Конфигурационное управление программного обеспечения является важной составляющей процесса разработки программного обеспечения. Тем не менее, несмотря на то, что конфигурационное управление программного обеспечения облегчает работу и страхует от многих неприятностей, зачастую очень сложно найти обоснование для инвестирования в соответствующие инструменты и процессы, которые не имеют прямой отдачи на первый взгляд. В данной работе создается модель, с помощью которой показан возврат от инвестиций в конфигурационное управление программного обеспечения, явно показываются затраты и выгоды.

The Abstract: Software configuration management is an important part of the software development process. Despite the fact that software configuration management makes work easier and ensures from many problems, it is often hard to find a reason for appropriate tools and processes investment, because there is no obvious benefits from such investment. This article describes a model that estimates the return of investments from software configuration management with direct costs and benefits.

Ключевые слова: конфигурационное управление программного обеспечения, программные метрики, измерение, окупаемость инвестиций.

Keywords: software configuration management, software metrics, measurement, return of investments.

Введение

Разработка и поддержка программного обеспечения (ПО) требуют взаимодействия многих дисциплин и групп людей. В модели СММ (Capability Maturity Model – модель технологической зрелости организации) [2, 3] Конфигурационное управление (КУ) ПО является одним из ключевых процессов, который необходим, чтобы подняться от первого «начального» уровня до второго – «повторяемого».

СММ может служить примером того, как КУ ПО повышает качество ПО и снижает затраты. Просто переход с первого до второго уровня модели СММ уже дает значительные преимущества. В отчете SEI (Software Engineering Institute) [4, 8] собраны данные по более чем 1233 отдельным проектам в 261 организации, расположенных в 10 странах, и приведена оценка выгод от достижения высоких уровней зрелости (Таблица 1).

Таблица 1

Пример выгод от использования КУ ПО

Уровень зрелости СММ	Календарные месяцы	Работа (рабочие месяцы)	Найденные дефекты	Ненайденные дефекты	Общая стоимость
1	29,8	593,5	1348	61	\$ 5 440 000
2	18,5	143,0	328	12	\$ 1 311 000
3	15,2	79,5	182	7	\$ 728 000
4	12,5	12,8	97	5	\$ 392 000
5	9,0	16,0	37	1	\$ 146 000

В таблице 1 показано, что при переходе с первого уровня зрелости на второй происходит сокращение времени выхода на рынок на 38% (столбец «Календарные месяцы»), сокращение найденных дефектов на 76%. В то же самое время стоимость второго уровня обходится в 25% от стоимости первого. Приведенный пример показывает значительные преимущества от применения КУ ПО.

В отличие от других ключевых процессов, таких как, управление требованиями, КУ ПО не является дисциплиной, которая на выходе дает ясный, реальный продукт, стоимость и качество которого можно измерить и оценить. КУ ПО на данный момент является прозрачной дисциплиной, жизненно необходимой для помощи в работе с другими дисциплинами. Большинство затрат и выгод КУ ПО «невидимы» и распыляются среди различных процессов и статей затрат проекта. Не смотря на это, затраты и отдачи от процесса КУ ПО всегда связывают только с отделом КУ ПО в команде. Необходимо понять, как именно достигаются результаты от применения КУ ПО. Возникает задача исследования первоначальных и будущих инвестиций в КУ ПО.

Непосредственно по предмету исследования встречается очень небольшое количество научных работ. Одной из них является [7], в которой авторы анализируют внедрение инструмента КУ ПО ClearCase в организации, занимающейся разработкой ПО. Но в упомянутой работе не дается полная модель, а производятся замеры некоторых величин до и после применения процессов и инструментов КУ ПО. Многие производители инструментов КУ ПО предлагают рекламные материалы, в которых обычно описываются простейшие модели оценки окупаемости, но эти модели не являются полными, либо величины, которыми они управляют, не заслуживают доверия. Такие простейшие модели могут быть адекватными для оценки при первичном внедрении КУ ПО, но они не достаточны для всеобъемлющей оценки окупаемости во время обновления инструментов КУ ПО, или при переходе к другим процессам КУ.

Задачей в данной работе является более тщательный анализ затрат и выгод от вложений в КУ ПО для получения более полной картины этих затрат и выгод. В данной работе будет выполнено описание одной модели в ситуации, когда компания только внедряет КУ ПО, и когда обновляет инструменты и/или процессы КУ ПО. Единственную модель проще использовать для оценки окупаемости вложений как при внедрении КУ ПО, так и при частичных изменениях процессов КУ ПО.

Модель оценки окупаемости не является единственной и конечной целью работы. Более перспективным видится обеспечение менеджера КУ ПО механизмами по повседневному мониторингу и настройке инструментов и процессов КУ ПО, что позволит прогнозировать конкретные затраты и выгоды от КУ ПО. Следовательно, необходимо найти набор специфических метрик КУ ПО. Традиционно под метриками КУ ПО понимают информацию, предоставляемую инструментами и процессами КУ ПО, с помощью которой можно производить

мониторинг других процессов разработки ПО, но не самих процессов КУ. В данной работе модель оценки окупаемости инвестиций рассматривается как промежуточный результат, который в дальнейшем будет использован для нахождения специфических метрик КУ ПО.

Создание модели для оценки окупаемости инвестиций, а так же разработка специфических метрик и оценок КУ ПО, выполняется обычно с помощью применения теоретического анализа совместно с эмпирическим подтверждением теории. Однако, как это было выяснено в работе [7], перечисленное требует длительного периода времени, и эмпирическое подтверждение на момент написания работы находится на стадии выполнения. Поэтому в данной статье будет представлен только теоретический анализ. Далее будет представлена модель оценки окупаемости инвестиций и ее анализ. Затем, будет описано, как использовать на практике подтверждение модели с помощью метрик и оценок КУ ПО.

1 Анализ процесса конфигурационного управления программного обеспечения

Чтобы получить широкое представление о масштабах потенциальных затрат и прибылях, необходимо проанализировать КУ ПО с нескольких различных перспектив. Первая, и наиболее очевидная, это анализ затрат и прибылей с точки зрения четырех канонических деятельности КУ ПО, которые описаны в стандартах, таких, как ISO 10007:2004 [6] и ANSI/EIA-649 [1]: идентификация конфигурации, управление изменениями, отчетность о статусе конфигурации и аудит конфигурации. Чтобы получить лучшее представление о границах прибылей, необходимо расширить область анализа и сфокусироваться не только на фазе кодирования процесса разработки ПО, но и на всем жизненном цикле, от анализа требований до этапа сопровождения. В конечном счете будет рассмотрена деятельность людей, занятых задачами КУ, для того, чтобы получить полное представление о выгодах от КУ ПО и затратах на него.

Далее приведено краткое описание результатов анализа различных перспектив КУ ПО. После, как результат анализа, приведено описание модели оценки окупаемости.

1.1 Анализ деятельности конфигурационного управления программного обеспечения

В общем случае, КУ ПО можно рассматривать как управленческую дисциплину, которая помогает в планировании и работе над проектом. В качестве таковой она предоставляет план КУ ПО, описывающий процессы, которым необходимо следовать, и инструменты (или набор инструментов), которые до некоторой степени могут автоматизировать некоторые из этих процессов. Общие стоимости складываются из стоимости инструментария (лицензии, поддержка), стоимости создания и поддержки плана КУ ПО, и стоимости обучения людей, чтобы они понимали и работали с установленными процессами инструментарием. Рассмотрим каждую из четырех деятельностей КУ ПО более подробно.

Идентификация конфигурации. Данная деятельность связана с фиксированием и распространением информации об идентифицируемых объектах конфигурации и фиксированием базовой конфигурации этих объектов [6]. Она устанавливает ясные и эффективные правила именования и правила иерархического структурирования объектов конфигурации. Это позволяет получать информацию с помощью понятной навигационной структуры и однозначную адресацию в различных контекстах.

Управление изменениями. Наиболее важное в этой деятельности – это процесс управления изменениями и совет управления изменениями (CCB – Configuration Control Board), которые определяют, как именно контролировать изменения. Это гарантирует, что результат от внесения изменений всегда оценивается и анализируется, и что сами изменения

всегда планируются и контролируются. Это позволяет отслеживать статус изменений так же, как и всего продукта в целом, через последовательность базовых конфигураций [1].

Отчетность о статусе конфигурации. Представляет собой систему форматированных отчетов, которые создаются на основе данных, имеющихся в системе КУ ПО. Связанные с данной деятельностью затраты – это создание исходных данных для отчетов. Полученные выгоды – это отслеживаемость нужной информации и наличие полной информации для управления.

Аудит конфигурации. Аудит проверяет функциональные характеристики, форму и соответствие продукта установленным требованиям. Затраты – это затраты на выполнение аудита, выгоды – это более стабильные и качественные базовые конфигурации и релизы.

1.2 Анализ жизненного цикла продукта

Обычно КУ ПО рассматривается как деятельность, нацеленная только на фазу кодирования проектов по разработке ПО. Однако, все фазы могут и должны применять принципы и методы КУ к своим рабочим продуктам. Рассмотрим каждую фазу жизненного цикла отдельно.

Постановка требований. Цель данного этапа работы над проектом заключается в создании описания того, что система должна делать, а также описания ее свойств и ограничений, например, таких, как операционная система, на которой должно запускаться разрабатываемое ПО, или язык реализации ПО. Применяя КУ ПО к фазе создания требований, можно получить те же прибыли (затраты), что и для фазы кодирования. Более того, можно получить трассируемость результатов работ на различных фазах.

Проектирование. На этапе проектирования создается архитектура системы. Данная фаза начинается с документа с требованиями, переходящего с предыдущей фазы, и переводит требования в архитектуру. Архитектура определяет компоненты, их интерфейсы и поведение.

Написание программного кода. В данной и следующих трех фазах можно получить наибольшее количество выгод от применения КУ ПО. Затраты данной фазы представляют собой затраты на инструментарий, процессы и обучение персонала. Выгодами является возможность управлять изменениями, повторно использовать код, работать параллельно (и/или удаленно), проводить автоматические сборки и иметь возможность устанавливать взаимодействие между программистами и командами и управлять ими.

Тестирование. Тестировщики получают преимущество от использования документированных базовых конфигураций, из которых можно создавать сборки проекта для тестирования. Это дает ясное представление о том, что именно тестируется в данный момент.

Релиз. Использование документированных базовых конфигураций позволяет всегда восстанавливать старые релизы. Более того, аудит конфигурации всегда гарантирует качество релиза.

Сопровождение. Данная фаза получает больше всего выгод от использования КУ ПО. Из информационной базы, предоставляемой КУ ПО, можно получать трассировку данных и историю событий, которые позволяют находить и устранять ошибки в ПО. Выгоды при добавлении новой функциональности такие же, как и для фазы кодирования.

1.3 Анализ сотрудников

Существуют различные роли, имеющие отношение к КУ ПО, и эти роли имеют различные интересы, как это указано в [5].

Высший менеджмент. Заинтересован в том, что происходит в процессе разработки продуктов, особенно в качестве создаваемых продуктов и уровне обслуживания клиентов. Больше всего высший менеджмент заботится о финансовых выгодах и затратах. Использование КУ ПО обеспечивает конкурентное преимущество в выпуске продукта на рынок раньше остальных, или помогает сократить время ответа на запрос конечного пользователя.

Менеджер проекта. Большинство выгод, которые интересуют данную роль, это планирование, и управление работами. Отчет о статусе помогает лучше понимать состояние проекта, а использование контролирующего органа гарантирует, что все изменения управляются, а воздействия анализируются, и все может быть завершено в установленные сроки и в установленный бюджет.

Менеджер КУ ПО. Является ответственным за разработку и применение процесса идентификации конфигурации, общепринятых соглашений именования и обозначения, управление базовыми конфигурациями. Сам факт, что такая роль существует, уже говорит о том, что существуют затраты на процесс КУ ПО. Кто-то должен отвечать за планы и процессы КУ ПО. Обычно менеджер КУ ПО так же принимает участие в собраниях контролирующего органа в качестве секретаря, и принимает участие в работе команды аудита.

Разработчик. Труд данной роли более всего подвержен воздействию внедрения процесса КУ ПО, поскольку этот процесс кардинально меняет работу программистов. Именно поэтому они могут сильно препятствовать внедрению новой дисциплины в их рабочий процесс. Поэтому для разработчиков необходимо организовывать обучение, чтобы они изучали новую дисциплину и уверенно себя чувствовали. Здесь термин «разработчик» используется в очень широком смысле, поскольку люди, участвующие в других фазах жизненного цикла могут использовать КУ ПО для «разработки» своих продуктов.

1.4 Результаты анализа

Из предыдущих трех разделов видно, что существует множество параметров и аспектов, затрагивающих внедрение дисциплины КУ ПО в процесс разработки ПО. Результат анализа приведен на рисунке 1.

Сложно представить выполнение любых задач КУ без использования соответствующих инструментов КУ ПО. Поэтому граф на рисунке 1 представлен с двумя начальными вершинами, которые представляют собой два вероятных источника выгод и затрат: дисциплина КУ ПО и инструментарий КУ ПО.

Дисциплина КУ ПО связана с выгодами верхней (руководящей) части иерархии компании. Дисциплина КУ ПО предназначена для описания организации рабочих групп, обеспечивая большую видимость проекта, и облегчая управление проектом.

Инструментарий КУ ПО связан с рабочим процессом разработчиков. Программисты получают поддержку параллельной работы и стабильную рабочую среду. Инструментарий КУ ПО повышает уровень обмена информацией между разработчиками, увеличивая их знания о проекте в целом. Инструменты автоматического слияния программного кода, полученного в результате параллельной работы, увеличивают скорость производства ПО.

1.5 Измеримость процесса КУ ПО

Чтобы определить степень измеримости параметров процесса КУ ПО, нужно рассмотреть два аспекта: объективность выгод с затратами, и количественная измеримость выгод с затратами. Согласно двум подходам, параметры процесса КУ ПО можно разделить на четыре региона (рисунок 2).

Смысл такого разделения параметров заключается в следующем.

Измеримые параметры могут использоваться в общем случае, для базовой модели оценки окупаемости, когда значения параметров можно взять из общих эталонных тестов, и они будут меняться очень незначительно от компании к компании.

Частично измеримые параметры не применимы ко всем компаниям и будут иметь величины, которые могут значительно меняться от компании к компании и, следовательно, должны оцениваться каждой компанией отдельно.

Неизмеримые параметры должны рассматриваться дополнительно от общего результата модели оценки окупаемости, они не применимы ко всем компаниям и пытаться оценить значения для этих параметров не имеет смысла.

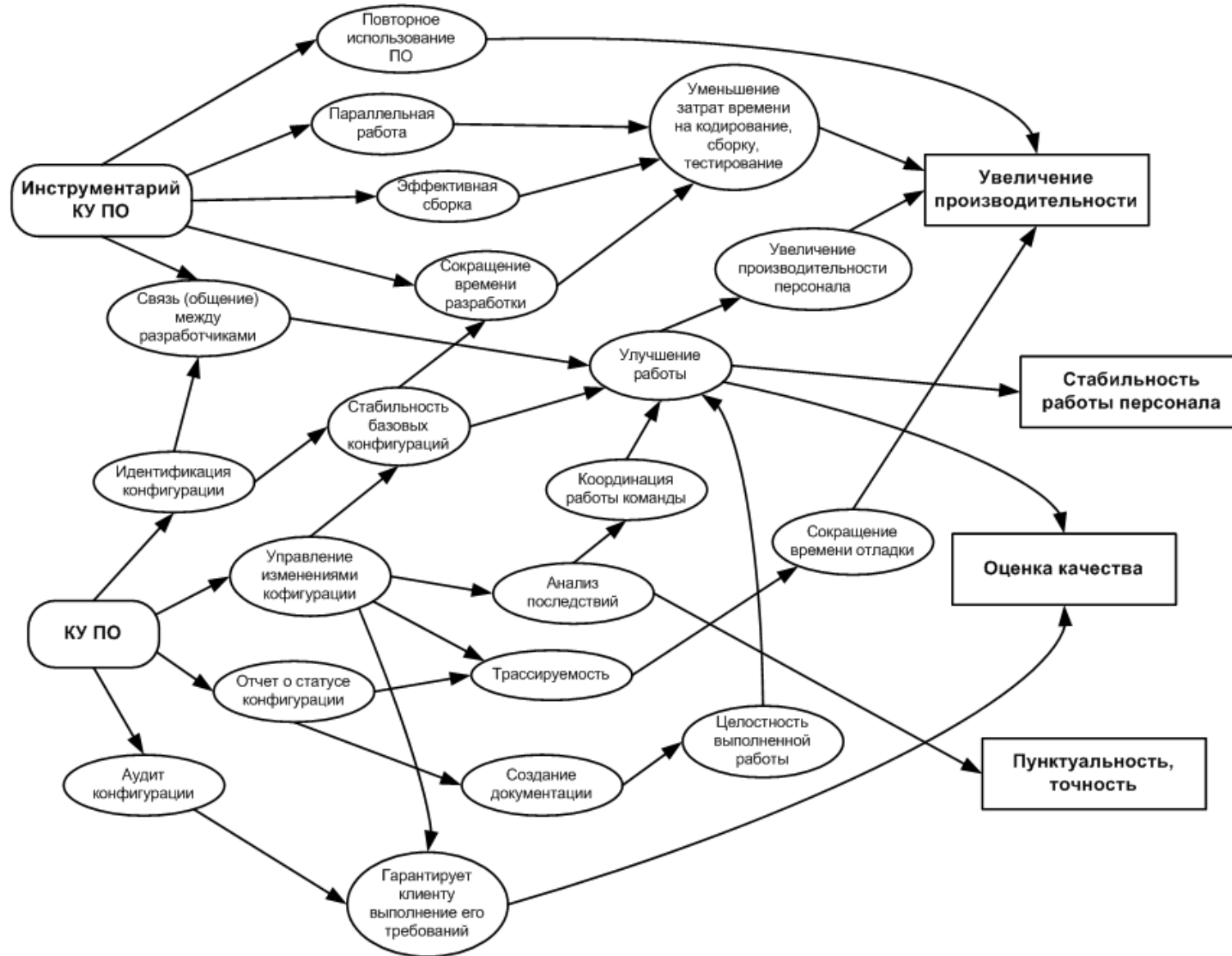


Рис. 1. Выгоды КУ ПО

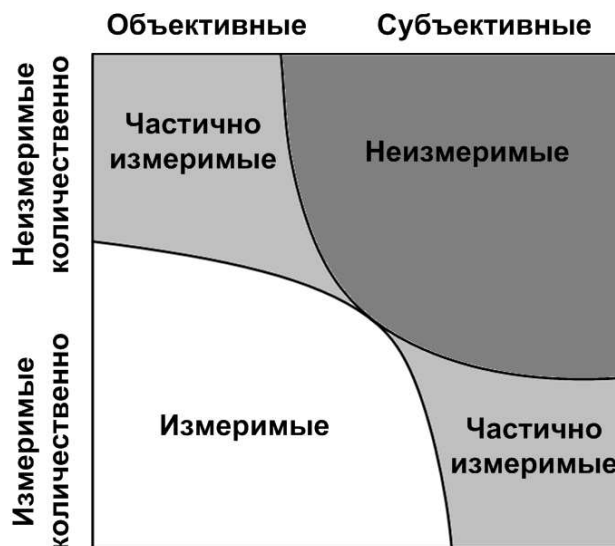


Рис. 2. Описание измеримости параметров

2. Итоговая модель оценки окупаемости инвестиций в КУ ПО

КУ ПО представляет собой дисциплину, зависящую от контекста применения. Возможные выгоды и соответствующая им рентабельность могут меняться от случая к случаю. Именно поэтому невозможно создать конечную формулу, применимую ко всем возможным случаям.

КУ ПО является субъективной дисциплиной, и поэтому в данной работе создана субъективная модель оценки окупаемости инвестиций, которая должна вычисляться компанией самостоятельно, учитывая контекст и способ применения КУ ПО.

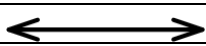
Затраты и выгоды делятся на три уровня измеримости. Они представлены в таблице 2.

Для модели оценки окупаемости инвестиций наличие большого количества неизмеримых параметров не является хорошим показателем. Но модель в данной работе является предварительной, поэтому в будущем, возможно, удастся передвинуть все параметры в область измеримых, пока будет происходить практическая работа с моделью. Более того, как можно заметить, для затрат большинство параметров можно измерить, в то время как для выгод большинство параметров неизмеримы. Это означает, что если использовать только измеримые и частично измеримые параметры для подсчета возврата инвестиций, то можно быть уверенным в том, что не стоит ожидать сюрпризов от «скрытых» затрат и можно ожидать еще больше прибылей в дополнение к просчитанной оценке окупаемости.

Что больше всего усложняет модель, так это то, что некоторые затраты и прибыли являются одноразовыми (например, покупка инструментария), или бывают один раз в год (например, оплата лицензий), в то время, как другие могут быть ежедневными (например, поддержка параллельной работы). Некоторые могут быть нелинейными (например, выгоды могут быть низкими до тех пор, пока не будет приобретен какой-либо инструментарий, или внедрен процесс). Поэтому довольно сложно сделать точные вычисления, учитывая все эти факторы. Но в источнике [7] сказано, что окупаемость для КУ ПО достаточно высока, чтобы не производить такие точные вычисления во время внедрения данного процесса. Когда придет время обновлять инструменты и/или процессы, то число соответствующих параметров вероятней всего будет достаточно низким, чтобы проводить точные вычисления. Однако в большинстве случаев наиболее интересным будет являться не вычисление выгодности КУ ПО, а оценка КУ ПО для конкретного проекта уже после того, как инструмент (инструменты) и процессы уже внедрены в стандарты компании.

Таблица 2

Модель оценки окупаемости

	Измеримые	Частично измеримые	Неизмеримые
Затраты	<p>1. Стоимость инструментов.</p> <p>2. Стоимость лицензий и технической поддержки.</p> <p>3. Стоимость обучения (стоимость обучения системного администратора и разработчиков работе с инструментарием и процессами КУ ПО).</p> <p>4. Дополнительная работа, связанная с задачами КУ ПО для:</p> <ul style="list-style-type: none"> – Менеджера КУ ПО; – Администратора; – Технического разработчика. 	<p>1. Процесс внесения изменения может быть гораздо сложнее (потери времени и денег в ожидании авторизации, время работы контролирующего органа).</p> <p>2. Потери времени на выполнение отчетов о статусе (зависит от инструмента и степени автоматизации).</p>	<p>1. Боязнь новых процедур.</p>
Прибыли	<p>1. Снижение внешних докладов о дефектах (частота отчетов о дефектах).</p> <p>2. Меньшее время на написание исправлений (bugfix).</p> <p>3. Возможность трассировки оригинального продукта на протяжении всего цикла разработки.</p> <p>4. Экономия времени с помощью автоматизированных сборок ПО.</p> <p>5. Управление версиями, параллельной работой, автоматическими слияниями.</p> <p>6. Трассируемость позволяет тратить меньше времени на верификацию, аттестацию и тестирование.</p>	<p>1. Сокращается количество изменений в численности персонала, облегчается интеграция новых работников (меньшая стоимость обучения новых работников).</p> <p>2. Позволяет управлять очень сложными деятельностью.</p> <p>3. Повторное использование существующего кода и сокращение усилий по повторной разработке.</p> <p>4. Увеличивающееся количество исправленных ошибок в разных версиях кода.</p> <p>5. Улучшенное сопровождение.</p> <p>6. Изменения планируются заранее, их воздействие заранее оценивается.</p> <p>7. Сокращение количества ошибок.</p>	<p>1. Работники более счастливы.</p> <p>2. Обеспечение коммуникации и координации в группе.</p> <p>3. Удовольствие от стабильной работы с фиксированной конфигурацией и собственной рабочей области.</p> <p>4. Работа на дому и распределенная разработка.</p> <p>5. Возможность выпускать продукт раньше.</p> <p>6. Сокращение времени ответов на запросы пользователей.</p> <p>7. Гарантируется, что пользователь получит то, за что заплатил.</p> <p>8. Аудит в конце каждой фазы гарантирует устойчивость работы.</p> <p>9. Обеспечивается доступность проекта.</p> <p>10. Достигается чувство организованности и контроля вместо хаоса.</p>
	Количественные		Качественные

Окупаемость инвестиций вычисляется для того, чтобы определить стоит ли вкладывать средства. Формула оценки окупаемости так же предполагает вычисление времени, которое потребуется организации на возвращение инвестиций, вложенных в КУ ПО.

Период окупаемости оценивается как точка пересечения на графике между кривой затрат и кривой выгод во времени (рисунок 3). Она находится решением системы уравнений из функции общих затрат ($Затр_{общ}$) и общих выгод ($Выг_{общ}$):

$$\begin{cases} Затр_{общ} = НЗ + \sum_i \left(\frac{Затр}{Месяц} \right)_i \\ Выг_{общ} = \sum_i \left(\frac{Выг}{Месяц} \right)_i \end{cases} \quad (1)$$

В (1) $Затр_{общ}$ – это функция общих затрат вложений, которая находится как сумма начальных затрат ($НЗ$), представляющих собой затраты на обучение на первых этапах внедрения и на покупку лицензий используемых инструментов, и повседневные затраты на поддержку системы КУ ПО. $Выг_{общ}$ – это сумма выгод, связанных с КУ ПО. Кривые функций, приведенные в (1), имеют вид, изображенный на рисунке 3.

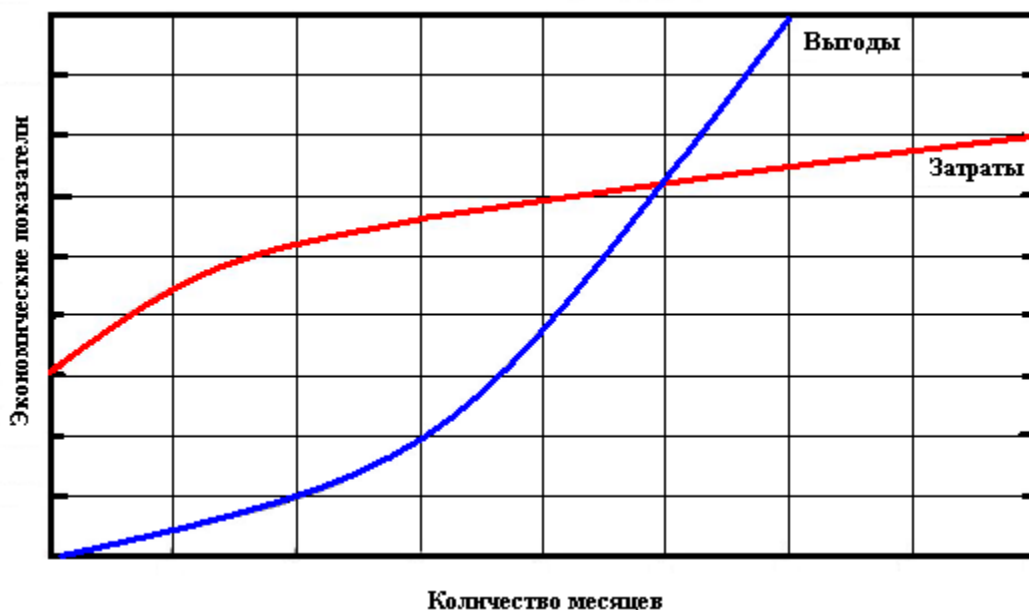


Рис. 3. Кривые выгод и затрат

Входными величинами для функций из (1) являются элементы первого столбца таблицы 2. Эти параметры используются, поскольку они не зависят от организации, в которой происходит расчет окупаемости инвестиций, и не зависят от того, как используется КУ ПО. В данном случае окупаемость инвестиций рассчитывается по общей формуле. С помощью предлагаемой модели возможно вычислить окупаемость инвестиций и по двум другим столбцам таблицы 2, но после некоторых дополнительных преобразований.

Общие затраты делят на три части: затраты на инструментарий (стоимость самого инструментария и лицензий на него), затраты на обучение и затраты на оплату дополнительной

работы персонала, связанной с внедрением процесса КУ ПО. Общие выгоды подразделяются на выгоды от меньшего количества исправлений (меньшее количество дефектов и сокращение времени на исправление каждого дефекта), уменьшения времени сборки продукта, управления версиями (параллельная работа и автоматическое слияние), сокращения времени тестирования. Учитывая сказанное, система (1) станет выглядеть следующим образом:

$$\begin{cases} \text{Затр}_{\text{общ}} = \text{Затр}_{\text{инст}} + \text{Затр}_{\text{обуч}} + \text{Затр}_{\text{рабКУПО}} \\ \text{Выг}_{\text{общ}} = \text{Выг}_{\text{деф}} + \text{Выг}_{\text{сбор}} + \text{Выг}_{\text{верс}} + \text{Выг}_{\text{тест}} \end{cases} \quad (2)$$

2.1 Затраты

Затраты на инструментарий, лицензии, поддержку. Данные элементы формулы хорошо измеряются. Общая формула будет следующей:

$$\text{Затр}_{\text{инст}} = N_{\text{польз}} \cdot \text{Лиц} \cdot \text{Лиц}_{\text{совм}} + \text{Затр}_{\text{сопр}}, \quad (3)$$

где $N_{\text{польз}}$ – количество пользователей, для которых покупается инструментарий, Лиц – стоимость одной лицензии, $\text{Лиц}_{\text{совм}}$ – степень совместного использования лицензии (возможность использовать одну лицензию несколькими пользователями), $\text{Затр}_{\text{сопр}}$ – стоимость сопровождения инструментария.

Затраты на обучение. Данные затраты зависят от используемого инструмента и от того, как именно он используется. Формула для данного вида затрат:

$$\text{Затр}_{\text{обуч}} = (\text{Раб}_{\text{адм}} + \text{Раб}_{\text{учит}}) \cdot \text{Ч}_{\text{адм}} + (N_{\text{польз}} \cdot \text{Раб}_{\text{польз}} + \text{Раб}_{\text{учит}}) \cdot \text{Ч}_{\text{польз}}, \quad (4)$$

где $\text{Раб}_{\text{адм}}$ – стоимость часа работы администратора системы КУ ПО, $\text{Раб}_{\text{учит}}$ – стоимость часа работы обучающего персонала, $\text{Ч}_{\text{адм}}$ – среднее время в часах, требуемое для обучения администратора системы КУ ПО, $\text{Раб}_{\text{польз}}$ – стоимость часа работы пользователя (разработчика) системы КУ ПО, $\text{Ч}_{\text{польз}}$ – среднее время в часах, требуемое для обучения пользователя (разработчика) системы КУ ПО.

К этим составляющим элементам формулы стоимости обучения можно добавить стоимость обучения нового персонала. Затраты (4) являются периодическими, поскольку в каждой организации увольняются старые и приходят на работу новые сотрудники. В случае высокой текучести кадров данные затраты можно так же подсчитать, если известна частота изменения персонала и средняя стоимость обучения одного человека процессу КУ ПО и пользования инструментарием КУ ПО.

Затраты на оплату дополнительной работы персонала по задачам КУ ПО. Менеджер КУ ПО тратит большую часть своего рабочего времени на идентификацию конфигурации [6], подготовку и участие в совещаниях совета управления изменениями, проведение аудита.

Общее время, потраченное на идентификацию конфигурации, можно разбить на временные промежутки, потраченные на работы по декомпозиции, управлению базовыми конфигурациями, созданию базовых конфигураций.

Время, потраченное на работу в качестве члена совета управления изменениями, может быть оценено как $N_{изм} \cdot T_{СУИ}$, где $N_{изм}$ – количество изменений в проекте за месяц, $T_{СУИ}$ – среднее время, затраченное на принятие решения советом управления изменениями.

Время, потраченное на аудит может быть оценено как среднее время, потраченное на один аудит ($T_{ауд}$), умноженное на среднее количество аудитов в месяц ($N_{ауд}$), то есть $T_{ауд} \cdot N_{ауд}$.

Чтобы получить общие затраты $Zатр_{рабКУПО}$, все потраченное время на КУ ПО умножается на стоимость работы персонала, который участвует в данном процессе, в единицу времени (час, день).

Остальной персонал, который тесно работает с инструментарием (разработчики, тестировщики), не добавляет существенных затрат к общей стоимости работ по КУ ПО, поэтому может не учитываться в общей формуле.

2.2 Выгоды

Здесь выгоды рассчитываются в рабочих часах, но они легко могут быть переведены в денежный эквивалент умножением количества часов на стоимость часа для конкретной категории работника, который приносит выгоды.

Меньшее количество исправлений (дефектов). Использование количества внутренних и внешних дефектов в месяц, и среднего времени на одно исправление (bugfix) до внедрения КУ ПО, а так же, знание степени сокращения количества исправлений и степени сокращения времени на одно исправление, позволяет оценить общее сэкономленное время за счет работы с дефектами.

$$Выг_{деф} = (\alpha_N \cdot \alpha_T \cdot N_{деф} \cdot T_{деф})_{внеш} + (\alpha_N \cdot \alpha_T \cdot N_{деф} \cdot T_{деф})_{внутр}, \quad (5)$$

где α_N – коэффициент сокращения количества дефектов, α_T – коэффициент сокращения времени, в среднем потраченного на один дефект, $N_{деф}$ – среднее количество дефектов в месяц, $T_{деф}$ – среднее время, потраченное на исправление одного дефекта.

Уменьшения времени сборки продукта после введения автоматической сборки. Окупаемость после внедрения автоматической сборки рассчитывается по формуле:

$$Выг_{сбор} = \sum_i \beta_i \cdot T_{сбор_i}, \quad (6)$$

где β_i – коэффициент сокращения объема работы, затраченный на i -ю сборку, $T_{сбор_i}$ – среднее время на i -ю сборку в проекте за месяц.

Управления версиями, параллельная работа и автоматическое слияние. Окупаемость от использования инструментария для управления версиями вычисляется по формуле:

$$Выг_{верс} = \sum_i \gamma_i \cdot T_{верс_i}, \quad (7)$$

где $T_{верс_i}$ – время, потраченное за месяц на создание, поиск и восстановление i -й версии, γ_i – коэффициент снижения объема работы с i -й версией.

КУ ПО позволяет выполнять параллельные работы, избегая классических проблем координирования работ, таких, как двойное сопровождение, проблемы совместного использования данных, одновременное обновление одного ресурса. Сокращение потерянного времени представлено в следующей формуле:

$$Выг_{верс2} = N_{пробл} \cdot T_{потер}, \quad (8)$$

где $T_{потер}$ – среднее потраченное время на решение одной классической проблемы координирования, $N_{пробл}$ – среднее количество таких проблем в месяц.

Улучшения, достигаемые за счет автоматических инструментов слияния, зависят от количества найденных различий между файлами, и от среднего количества операций слияния, выполняемых за месяц (количество новых версий, или количество новых ветвей). Зная среднее время, потраченное на одно слияние файлов ($T_{слиян}$), и среднее количество слияний в месяц ($N_{слиян}$), общие выгоды могут быть получены с помощью следующей формулы:

$$Выг_{верс3} = \gamma_{слиян} \cdot N_{слиян} \cdot T_{слиян}, \quad (9)$$

где $\gamma_{слиян}$ – коэффициент сокращения объема работы.

Сокращения времени тестирования. Зная среднее время за месяц, потраченное выполнения одного теста ($T_{тесті}$), и скорость сокращения этого времени для конкретного теста (δ_i), можно получить формулу для оценки прибыли от сокращения времени тестирования:

$$Выг_{тест} = \sum_i \delta_i \cdot T_{тесті}. \quad (10)$$

2.3 Алгоритм оценки окупаемости инвестиций от применения КУ ПО

Во время первых этапов внедрения процесса КУ ПО коэффициент усиления, связанный с элементами модели может меняться, поскольку в данный период персонал изучает новые процессы, и основы работы с инструментарием КУ ПО. По этой причине окупаемость инвестиций должна вычисляться с учетом ежемесячных затрат и выгод. Эти вычисления выполнены с помощью итерационного алгоритма, в котором изменения коэффициента усиления учитываются за каждый период (рисунок 4).

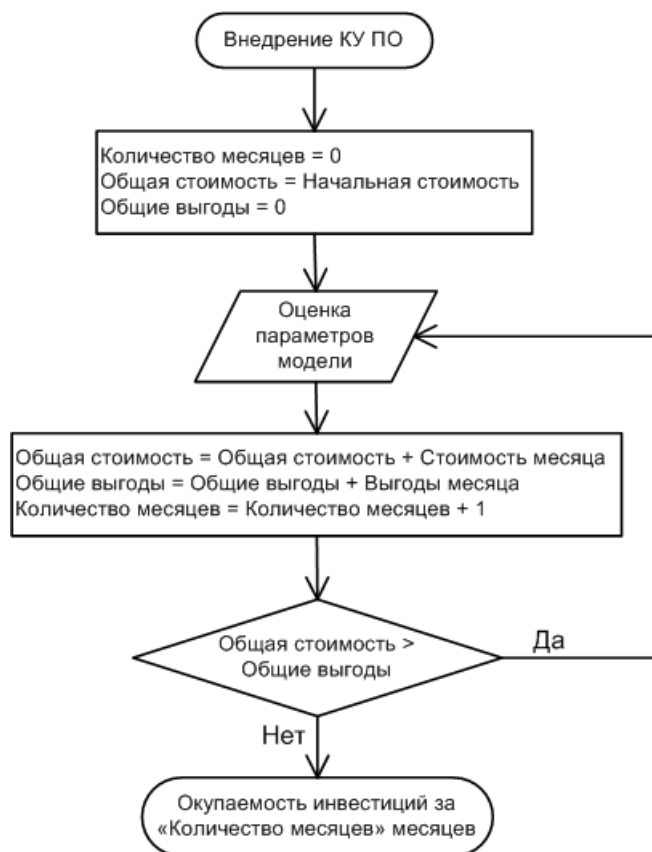


Рис. 4. Алгоритм оценки окупаемости инвестиций

3 Дальнейшие работы

Одним из дальнейших направлений исследований может стать более точная оценка коэффициентов усиления, которые присутствуют в формулах модели оценки окупаемости. Данное направление можно реализовать практическим способом, оценивая работу компаний, активно применяющих процессы КУ ПО, и анализируя исторические данные их проектов. Такой анализ потребует значительного количества времени, поскольку сбор данных в объемах, достаточных для статистического анализа, займет порядка нескольких лет.

Другим направлением, развивающим результаты данной работы, будет являться практическое применение полученных теоретических выкладок. Практическое применение в организациях, использующих КУ ПО, поможет оценить эффективность и завершенность предложенной модели оценки окупаемости.

Уточнение модели оценки окупаемости так же является предметом дальнейших исследований. Составляющие ее элементы могут быть разбиты на более мелкие параметры, если после апробации выяснится неточность существующих элементов. Так же необходимо развивать методы оценок неизмеримых, или частично измеримых параметров.

Заключение

В данной работе сделана попытка оценить возможные выгоды и затраты, появляющиеся в результате внедрения КУ ПО в организации, занимающейся производством ПО. Во время проведенного анализа выяснилось, что количество общепринятых параметров необходимо расширить, чтобы учесть все аспекты КУ ПО, в том числе и те, которые не поддаются измерениям. Присутствие в модели множества субъективных параметров, связанных с КУ ПО, вы-

явило необходимость в разработке метода, который бы позволял количественно измерять плохо измеримые параметры.

По сравнению с некоторыми уже существующими работами, полученная модель более полная, касательно параметров затрат и выгод. Результатом дальнейших работ будет уточнение полученной модели и апробация ее на реальных проектах, к которым применяются процессы КУ ПО.

ЛИТЕРАТУРА

1. ANSI/EIA 649-B-2011 – Configuration Management Standard.
2. Capability Maturity Model Integration (CMMI). Режим доступа <http://www.sei.cmu.edu/cmmi/> (дата обращения 21.09.2012).
3. CMMI Institute - the home of Capability Maturity Model Integration. Режим доступа <http://cmmiinstitute.com/> (дата обращения 10.12.2012).
4. Dave Kitson, Steve Masters. An Analysis of SEI Software Process. Assessment Results 1987-1991 // TECHNICAL REPORT, 1992, pp. 35. Режим доступа <ftp://ftp.sei.cmu.edu/pub/documents/92.reports/pdf/tr24.92.pdf> (дата обращения 04.12.2012).
5. Farah Joe. Metrics and Process Maturity // The Configuration Management Journal, December 2004.
6. ISO 10007:2003 Quality management systems – Guidelines for configuration management.
7. Larsen J.O., Roald H.M. Introducing ClearCase as a Process Improvement Experiment // Proc. of the 8-th International Symposium on System Configuration Management, London, UK, 1998.
8. Merant: The Business Case for Software Configuration Management // White Paper, 2001. Режим доступа http://www.synergex.com/pdf/Business_Case_SCM.pdf (дата обращения 06.08.2012).