

Розглянуто три основні концепції оцінювання ризиків, що найбільш розповсюджені на практиці. Концепція ризику як можливості. Концепція ризику як небезпеки. Концепція невизначеності. Кожна з цих стратегій представляє певний інтерес, оскільки застосування першої концепції пов'язано з отриманням певної вигоди від функціонування системи, другої - з мінімізацією наслідків негативних впливів та

максимізацією позитивних, а третьої - з поліпшенням передбачуваності поведінки системи.

Розглянуто узагальнену модель оцінювання ризиків, ймовірнісну природу ризику.

Особливу увагу приділено аналітичним методам аналізу та управлінню інформаційними ризиками, що здійснюються безпосередньо на основі значень екзогенних змінних.

Література

1. Балдин К.В. Управление рисками и выбор стратегии [Текст] / К.В. Балдин // Высшее образование сегодня. - Реферируемое издание ВАК России. - Б.м. - 2005.- № 11. - С. 36-38.
2. Риски систем: оценка и управление [Текст] / В.Н. Асеев, Д.Е. Морев, В.Б. Щербаков. - под. ред. Ю.Н. Лаврухина - Воронеж: Междунар. ин-т компьют. технологий, 2007. - 261 с.
3. Вишняков Я.Д. Общая теория рисков: учеб. пособие для студ. высш. учеб. Заведений [Текст] / Я.Д. Вишняков, Н.Н. Радаев. - М.: Издательский центр «Академия», 2007. - 368 с.
4. Петренко С.А. Управление информационными рисками. Экономически оправданная безопасность [Текст] / С.А. Петренко. - М.: Компания АйТи; ДМК Пресс, 2004. - 384 с.
5. Черныш В.И. Методы оценивания информационных рисков компании [Текст] / В.И.Черныш // Материалы XV Международного юбилейного молодёжного форума «Радиоэлектроника и молодежь в XXI веке»: Сб. тезисов, 18-20 апреля 2011 р., Т.5. - Харьков: ХНУРЭ. 2011. - С. 195.
6. Замула О.А. Аналіз міжнародних стандартів в галузі оцінювання ризиків інформаційної безпеки [Текст] / О.А. Замула, В.І. Черныш // Системи обробки інформації. - Харків: ХУ ПС, 2011. - Вип.2(92). - С.53-56.
7. Замула А.А. Оценка рисков информационной безопасности в современных информационных системах [Текст] / А.А. Замула, В.И. Черныш, К.И. Иванов // XIV Международная научно-практическая конференция «Безопасность информации в информационно – телекоммуникационных системах», тезисы докладов. - К.: ЧП «ЕКМО», НИЦ «ТЕЗИС» НТУУ «КПИ», 2011. - С. 31.

Запропоновано сформулювати багатокритеріальну задачу досягнення заданого рівня зрілості процесу розробки програмного забезпечення на основі вибору серед альтернативних варіантів реалізації цього процесу з урахуванням ресурсних обмежень організації

Ключові слова: поліпшення якості, модель зрілості

В работе предложено сформулировать многокритериальную задачу достижения заданного уровня зрелости процесса разработки программного обеспечения на основе выбора среди альтернативных вариантов реализации данного процесса с учетом ресурсных ограничений организации

Ключевые слова: улучшение качества, модель зрелости

We propose to formulate the problem of achieving the necessary level of maturity for the software process based on the selection among alternate variants of the process steps implementation under the resource constraints of the organization

Key words: quality improvement, maturity model

УДК 681.518

ОЦЕНКА И УПРАВЛЕНИЕ КАЧЕСТВОМ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ МОДЕЛЕЙ ЗРЕЛОСТИ

В. А. Шеховцов

кандидат технических наук, доцент*

Контактный тел.: (050) 9653427

E-mail: shekvl@yahoo.com

М. Д. Годлевский

Доктор технических наук, профессор, заведующий

кафедрой*

И. Л. Брагинский

Аспирант

*Кафедра автоматизированных систем управления

НТУ «ХПИ»

ул. Фрунзе 21 Харьков, Украина, 61002

1. Введение

Важность IT-индустрии в масштабах Украины в последние годы существенно возросла. Фактически повышение продуктивности всей отрасли может иметь позитивное влияние на общество в целом: чем больше проектов по разработке ПО завершаются успешно, тем выше будет уровень зарплат IT-специалистов, что, в свою очередь, приведет к росту финансовых поступлений в бюджет страны в виде налогов или поддержки отечественных производителей путем приобретения их товаров или услуг.

Проблема в данном случае заключается в том, что это влияние может быть реализовано только в том случае, если проекты, выполняемые отечественными компаниями, будут успешными. Как следствие, факторы, влияющие на успех проектов по разработке ПО, можно считать важными для Украины – как с социальной, так и с экономической точки зрения. В данной работе рассматривается общая проблема улучшения качества процесса разработки программного обеспечения (ПРПО).

Работа имеет следующую структуру. Разделы 2 и 3 содержат обзор существующих подходов к описанию жизненного цикла программного обеспечения. Раздел 4 посвящен методам улучшения качества процесса разработки программного обеспечения. В разделе 5 перечисляются проблемы современных методов исследования и улучшения качества процесса разработки программного обеспечения, формулируется постановка задачи управления качеством ПРПО и дается обзор подходов к ее решению.

2. Процесс разработки программного обеспечения

Первым исследователем, который обратил внимание на необходимость реализации процессного подхода к разработке ПО, был Хамфри, основные результаты работ которого отражены в монографии [1]. В работе [2] им было приведено следующее определение: процесс разработки ПО – это весь набор действий, относящихся к программной инженерии, необходимый для того, чтобы преобразовать пользовательские требования в ПО.

Документ SWEBOOK [3], описывающий базовый набор знаний по программной инженерии; также известный как стандарт ISO/IEC 19759, приводит следующее определение: под ПРПО понимается набор действий, методов, практических процедур и преобразований, используемых людьми для разработки и поддержки программного обеспечения и связанных с ним продуктов.

Лаврищева [4] определяет ПРПО (под названием «базовый процесс программной инженерии») как множество логично связанных видов организационной деятельности организации-разработчика и набора средств и инструментов, относящихся к изготовлению программного продукта.

Как можно увидеть, приведенные определения различаются только частностями, будучи согласованными в основном. Во всех определениях под ПРПО понимается упорядоченная совокупность действий, ставящих своей целью создание программного про-

дукта. Различия фактически связаны с интерпретацией конкретного набора действий.

3. Процессы и модели жизненного цикла программного обеспечения

В данном разделе мы остановимся на описании структуры конкретного экземпляра ПРПО и набора его элементов. Такие описания определяют набор этапов ПРПО, порядок следования этих этапов при реализации конкретных проектов и критерии перехода от этапа к этапу. Фактически они задают определенные подходы к реализации ПРПО.

В данном случае требуется получить ответ на два основных вопроса [5].

1. Какой набор этапов ПРПО используется в конкретной реализации?

2. Каким образом задается порядок следования этих этапов и условия перехода между ними?

В современной литературе эти вопросы, как правило, рассматриваются по отдельности. Ответ на первый вопрос дает определение процессов жизненного цикла (ЖЦ) ПРПО (software lifecycle processes; каждый такой процесс выполняется на одном или нескольких этапах ЖЦ ПРПО. Для краткости мы будем использовать для этих процессов сокращение ПЖЦ. Ответ на второй вопрос дает определение модели жизненного цикла (ЖЦ) ПРПО (МЖЦ); такая модель описывает последовательность выполнения ПЖЦ и условия перехода между ними.

Процесс разработки программного обеспечения представляется в виде организованной совокупности ПЖЦ, каждый из которых отвечает за определенную задачу, выполняемую при разработке. Фактически такие процессы могут рассматриваться как подпроцессы ПРПО. Существует множество вариантов набора ПЖЦ, используемых в конкретных реализациях процесса разработки. Как правило, такие варианты содержат множество общих элементов. В работе [6] приведен общий набор ПЖЦ, являющийся покрытием большинства существующих вариантов.

1. Системная инициализация (планирование).
2. Анализ и задание требований.
3. Функциональная спецификация и задание прототипов системы.
4. Декомпозиция ПО на компоненты и выбор стратегии их реализации.
5. Архитектурное и детальное проектирование.
6. Реализация и отладка компонентов.
7. Интеграция и тестирование программного обеспечения.
8. Создание документации и поставка ПО заказчику.
9. Развертывание ПО на стороне заказчика.
10. Обучение персонала работе с ПО.
11. Поддержка развернутого ПО.

Можно привести ряд наборов ПЖЦ, используемых на практике.

1. Набор ПЖЦ, определенный стандартом ISO/IEC 12207: «Системная и программная инженерия. Процессы жизненного цикла программного обеспечения» [7].

2. Набор ПЖЦ, определенный стандартом SWEBOK – Software Engineering Body of Knowledge (стандарт ISO/IEC19759) [3].

3. Набор ПЖЦ (именуемых деятельностями), определенный методом улучшения процесса разработки, основанным на модели зрелости CMMI.

4. Набор ПЖЦ, определенный методом улучшения процесса разработки, основанным на модели зрелости SPICE (заданной стандартом ISO/IEC 15504) [8].

К наиболее распространенным МЖЦ относятся: модель водопада [9], спиральная модель [10], модель инкрементно-итеративной [11,12] и активной (agile) [11,13] разработки.

4. Улучшение качества процесса разработки программного обеспечения

Под улучшением процесса разработки программного обеспечения (Software Process Improvement, SPI) [1, 14, 15] понимается совокупность действий, направленная на улучшение характеристик процесса разработки в результате выполнения некоторого набора мероприятий. Важным моментом здесь является тот факт, что критериями успеха данных действий являются характеристики процесса разработки (его документированность, управляемость и т.д.), а не характеристики разрабатываемого ПО. В этом данная технология опирается на подходы инженерии качества, в том числе на тотальное управление качеством (Total Quality Management) [16, 17]. Предполагается, что улучшение характеристик процесса разработки напрямую связано с качеством его результатов.

Среди подходов к улучшению процесса разработки можно выделить подходы, основанные на стандарте ISO 9001:2000 [18], подходы, основанные на понятии модели зрелости ПО [8, 15, 19], подходы, основанные на технологии Six Sigma [20]. В данной работе мы остановимся более подробно на подходах, основанных на модели зрелости программного обеспечения.

4.1. Понятие модели зрелости ПО

Модель зрелости ПО является одним из основных понятий, связанных с улучшением процесса разработки [8, 19]. Отметим, что слово «модель» в данном названии является скорее данью традиции, так как данное понятие фактически описывает некоторый метод или технологию. Поэтому в дальнейшем будем говорить о технологии «модель зрелости» (ТМЗ).

ТМЗ представляет собой описание идеального движения организации (или отдельного подразделения) к желаемому уровню реализации процесса разработки, заданное в виде нескольких последовательных этапов (которые нужно выполнить) или уровней (которых нужно достичь), дополненное средствами оценивания полноты выполнения описанных этапов или соответствия процесса организации описанным уровням. Такая модель позволяет организации оценить свою реализацию процесса разработки путем ее сравнения с лучшими вариантами такой реализации (или с вариантами конкурентов). Цель такого сравнения

– сформировать структурированный план улучшения этой реализации ПРПО.

ТМЗ опирается на следующие понятия:

1. Фокусная область, представляющая собой категорию действий, связанных с отдельной дисциплиной процесса разработки (например, анализ требований, проектирование или конструирование ПО).

2. Практика, представляющая собой отдельное действие в рамках фокусной области (например, сбор требований у заинтересованных лиц).

Фокусные области и практики имеют прямую связь с элементами модели процесса разработки ПО. Фокусные области соответствуют, как правило, отдельным процессам ЖЦ ПО и включают в себя все действия, напрямую или косвенно связанные с таким процессом. Конкретные практики могут соответствовать как конкретным процессам ЖЦ ПО, так и отдельным их подпроцессам.

В настоящее время ТМЗ принято относить к одной из двух главных категорий: непрерывные и дискретные (поэтапные) [21].

Непрерывные ТМЗ описывают подход, при котором для каждой фокусной области задается набор уровней возможностей, которые определяются реализованными практиками. Такие уровни определяются для каждого конкретного проекта, а не для всей организации, при этом достигается большая гибкость (для разных проектов можно определять разные уровни возможностей), но оценивать всю организацию не получается (теоретически можно получить интегральную оценку на данный момент, взяв все текущие проекты, но такая оценка будет зависеть от времени).

Дискретные ТМЗ выделяют отдельные уровни зрелости для оценивания всей организации. Для каждого уровня задается набор фокусных областей (или практик в рамках отдельных фокусных областей), которые должны быть реализованы для того, чтобы организация оказалась на заданном уровне зрелости. Такие ТМЗ являются менее гибкими и не работают для отдельных проектов, но при этом позволяют оценивать организацию в целом и производить сравнение на уровне организаций.

4.2. CMMI как пример технологии «модель зрелости»

Рассмотрим важный пример практической реализации смешанной дискретно-непрерывной ТМЗ – Capability Maturity Model Integration – CMMI [19].

Сначала остановимся на особенностях использования CMMI как дискретной ТМЗ. CMMI определяет пять уровней зрелости, которые могут быть достигнуты организацией по итогам проверки на соответствие стандарту.

1. Начальный уровень. У организации мало или нет вовсе общих процессов. Успех проектов полностью зависит от усилий и опыта сотрудников. Большинство компаний находятся на этом уровне.

2. Повторяемый уровень. В организации внедрены стандартные процессы управления проектами, которые используются во всех проектах. Есть попытки создать базовый фундамент, на котором в будущем будет строиться дальнейшие улучшения. Большинство

компаний, начавших движение по пути СММІ, стараются достичь этого уровня.

3. **Определенный уровень.** В данном случае в организации внедрены стандартные процессы. Ведется работа по достижению стандартизации в производственной деятельности организации подобно тому, как это велось для процессов управления проектами на уровне 2. Это может включать единство технологий, инструментария, процедур, способов и т.д.

4. **Управляемый уровень.** В процессе разработки реализована обратная связь по управлению. Производятся измерения всех аспектов процессов управления проектами и производства. Ведется хранилище метрик и ключевых знаний, полученных в завершенных проектах.

5. **Оптимизирующийся уровень.** В данном случае организован замкнутый цикл исполнения процессов, измерений и непрерывного улучшения. При этом непрерывно используются измерения и обратная связь, а также творческие силы сотрудников для оптимизации процессов.

Фокусные области в СММІ называются процессными областями. Определено 22 таких области, в значительной степени соответствующих процессам ЖЦ ПО в ISO/IEC 12207.

Для каждой из процессных областей существует ряд целей, которые должны быть достигнуты при внедрении СММІ в данной процессной области. Некоторые цели являются уникальными — они называются специфическими. Общие цели применяются к нескольким процессным областям.

Цели достигаются при помощи реализации практик. В соответствии с отношением к какой-либо цели, практики делятся на специфические и общие. Понятие практики в СММІ полностью соответствует общему определению, приведенному ранее.

4.3. Исследования в области улучшения качества процесса разработки ПО

Прежде всего отметим, что ряд исследователей предлагает полноценные альтернативные реализации подходов к улучшению процесса разработки, не опирающиеся на понятие ТМЗ. В качестве примера такой технологии можно привести ProRAM [22].

Более широкое распространение получили подходы, опирающиеся на понятие ТМЗ. Например, в работе [23] предлагается универсальный подход к разработке непрерывных ТМЗ и приводятся примеры применения данного подхода к оцениванию архитектуры ПО и качества процесса управления программным продуктом. Работа [24] посвящена разработке ТМЗ для управления информационной инфраструктурой предприятия. В работе [25] предлагается создание «решеток зрелости» для оценивания организационных возможностей предприятия, каждая такая решетка дает возможность получить комплексную оценку возможностей одновременно на разных уровнях зрелости.

Предложен ряд моделей качества для оценивания и сравнения подходов к улучшению качества процесса разработки ПО. В частности [26] предлагает таксономию характеристик качества и набор метрик для сравнения решений в данной области, другой подход к

сравнению подходов предложен в [27]. В диссертации [28] предлагается расширенный набор критериев для оценивания решений по улучшению процесса разработки.

Ряд исследователей опубликовал реализации процедур оценивания процесса разработки на соответствие заданному уровню зрелости для конкретных ТМЗ [29]. Работа [30] предлагает «матрицу зрелости» для управления программными продуктами, которая содержит набор характеристик для оценивания качества процесса в IT-организации.

Ряд работ объединяет идеи синтеза и улучшения процесса разработки для получения гибких решений по улучшению процесса разработки. Например, подобное гибкое решение для реализации процесса жизненного цикла для этапа обработки требований приведено в [31].

В работе [32] предложено представлять подходы по улучшению процесса разработки, относящиеся к категории управления информационной службой предприятия, в виде фрагментов такого процесса и применять для сборки окончательного процесса методологию инженерии процесса разработки.

В работе [33] введено понятие ситуационного роста уровня зрелости процесса разработки. Вводится набор действий по повышению уровня зрелости в соответствии с некоторой моделью зрелости и ставится задача выбора оптимального набора действий для достижения необходимого уровня зрелости в случае дискретной модели. В то же время не приведен конкретный алгоритм такого выбора.

В работе [34] вводится понятие ситуационной ТМЗ, которая формируется для каждого конкретного случая отдельно. Предложен универсальный метод формирования таких ТМЗ.

5. Постановка задачи управления качеством процесса разработки ПО на основе технологии «модель зрелости»

Проведенный анализ существующих решений в области улучшения процесса разработки показал, что до настоящего времени данная область знаний не получила должной формализации, в частности:

а) до сих пор не предложено формальной модели уровня зрелости, фокусной области и практики;

б) недостаточно формализованы и исследованы факторы, влияющие на достижение зрелости ПРПО, факторы оценивания зрелости процесса разработки и факторы оценивания эффекта от внедрения технологии повышения зрелости процесса разработки, недостаточно изучено взаимовлияние факторов.

Кроме того, можно отметить, что традиционный подход к внедрению стандартных практик улучшения процесса разработки обладает рядом существенных недостатков.

Прежде всего, такие практики определяют некоторый набор значений характеристик процесса, которых необходимо достичь для того, чтобы процесс был сертифицирован, т.е. о нем сказано, что он соответствует некоторому уровню. Это важно, в частности, для получения контрактов.

При этом не детализируется, каким образом можно добиться целевых значений этих характеристик, описывается только результат и общие «наставления». Очевидно, что для достижения этих результатов можно использовать разные подходы, но в текущей литературе по предмету критерии выбора подходов не задаются. Фактически предоставляется некоторая библиотека практик без критериев, определяющих, как ей пользоваться. Нам известна только работа [35], в которой ставится задача принятия решений для выбора процессных областей, в которых предполагается выполнять набор работ с целью улучшения качества процесса разработки, но данная работа не предлагает конкретного алгоритма выбора, применимого как для дискретных, так и для непрерывных моделей зрелости.

Кроме того, недостаточно освещен вопрос зависимости степени достижения цели, связанной с сертифициацией организации в соответствии с заданной моделью зрелости, от ресурсов, которыми обладает организация, так, например, работа [35] не учитывает ресурсные ограничения вовсе.

Возникает задача построения формального алгоритма достижения заданного уровня зрелости процесса разработки программного обеспечения на основе выбора среди альтернативных вариантов реализации каждого из этапов данного процесса. Для этого необходимо:

1. Построить формальную модель стандартной практики, при этом за основу можно взять практики одной методологии улучшения процесса разработки, например, СММІ.
2. Построить формальную модель уровня зрелости.
3. Выделить критерии, определяющие измеримые характеристики процесса разработки, с одной стороны, и характеристики, влияющие на достижение определенного условия для некоторой практики, с другой стороны.
4. Провести синтез процесса разработки на основе этих критериев, при этом целью синтеза предлагается достижение полученным процессом разработки заданного и формализованного уровня зрелости.
5. При синтезе процесса разработки необходимо учитывать ресурсные ограничения с целью получения

процесса, который может быть внедрен в конкретной организации, учитывая ее ресурсные возможности.

Сформулированная задача обладает несколькими критериями, которые можно объединить в группы. В частности, к одной группе критериев относятся критерии, связанные с достижением заданного уровня зрелости, к другой – критерии, связанные с необходимостью эффективно использовать ресурсы, находящиеся в распоряжении организации-разработчика.

Для решения данной задачи предлагается задействовать методы многокритериальной оптимизации и динамического программирования.

6. Выводы и перспективы дальнейшей работы

В данной работе было рассмотрено современное состояние исследований в области оценки и управления качеством процесса разработки программного обеспечения.

По результатам данного обзора можно сделать вывод о серьезных недостатках современных подходов к управлению качеством процесса разработки программного обеспечения и о том, что формулировка задачи синтеза ПРПО с целью достижения заданного уровня зрелости и с учетом ресурсных ограничений организации-разработчика в настоящее время является актуальной

Дальнейшие исследования будут проводиться по следующему плану.

1. Разработка формального представления ТМЗ в целом, уровней зрелости, фокусных областей и практик, проверка этих ТМЗ на соответствие реальности.
2. Формальная постановка задачи управления синтезом процесса разработки программного обеспечения с учетом ограничений на ресурсы.
3. Реализация системы поддержки принятия решений, которая опирается на набор программных компонентов решения задачи управления синтезом процесса разработки программного обеспечения с учетом ограничений на ресурсы.
4. Проведение исследований применимости разработанных методов для проектов различной направленности.

Литература

1. Humphrey W.S. Managing the software process / W.S. Humphrey. – Boston, MA: Addison-Wesley. – 1989.
2. Humphrey W.S. The Software Engineering Process: Definition and Scope / W.S. Humphrey // ACM SIGSOFT Software Engineering Notes. – 1989. – Vol. 14. – N 4. – P. 82-83.
3. Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK), ISO/IEC TR 19759:2005. ISO. – 2005. – 173 p.
4. Лаврішча К.М. Програма інженерія / К.М. Лаврішча. – К: Академперіодика. – 2008. – 319 с.
5. Madachy R.J. Software process dynamics / R.J. Madachy. – Hoboken, NJ: IEEE Press - Wiley Inter-science. – 2008. – 601 p.
6. Scacchi W. Process models in software engineering / W. Scacchi // J.Marciniak (ed.) Encyclopedia of Software Engineering, 2nd edition. – New York: John Wiley & Sons. – 2001.
7. ISO/IEC 12207:2008, Information technology – Software life cycle processes. International Organization for Standardization. – 2008. – 122 p.
8. Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем. М: Книга. – 2001. – 348 с.
9. Boehm B. W. Software engineering / B.W. Boehm // IEEE Transactions on Computers. – 1976. – Vol. 25. – N 12. – P. 1226–1241.

10. Boehm B.W. A spiral model of software development and enhancement / B.W. Boehm // IEEE Software. – 1988. – Vol. 21. – N 5. – P. 61-72.
11. Larman K. Agile and Iterative Development: A Manager's Guide / C. Larman. – Addison Wesley –2003. – 368 p.
12. Крачтен Ф. Введение в Rational Unified Process / Ф. Крачтен. – К: Вильямс. – 2002. – 240 с.
13. Schwaber K. Agile Software Development with Scrum / K. Schwaber. – Prentice Hall,., – 2001. – 158 p.
14. Persse J.R. Process Improvement Essentials / J.R. Persse. – O'Reilly. – 2006.
15. Mutafelija, B. Process improvement with CMMI v1.2 and ISO standards / B. Mutafelija. – Auerbach Pubs. – 2009.
16. Шубенкова Е.В. Тотальное управление качеством / Е.В. Шубенкова. – М: XXI. – 2005. – 252 с.
17. E.Y. Li. Total Quality Management in Software Development Process / E.Y. Li, H.-G. Chen, W. Cheung // The Journal of Quality Assurance Institute. – 2000. – Vol. 14. – N 1. – P. 4-6, 35-41.
18. Schlickman J. ISO 9001:2000 Quality Management System Design / J. Schlickman. – Artech House. – 2003.
19. Chrissis M.B. CMMI: Guidelines for Process Integration and Product Improvement / M.B. Chrissis, M. Konrad, – Addison-Wesley. – 2003. – 688 p.
20. Motorola. Motorola University Six Sigma Arti-cles [Электронный ресурс] / Motorola. –. – 2006. – Режим доступа: <http://www.motorola.com>.
21. van Steenberg M. An Instrument for the Development of the Enterprise Architecture Practice / M. van Steenberg, M. van den Berg, S. Brinkkemper // Proc. ICEIS'2007. –, Vol. 3. – 2007. – P. 14-22.
22. P.V. Martins. ProPAM: SPI based on Process and Project Alignment / P.V. Martins, A.R.d. Silva // IRMA Internacional Conference. – 2007.
23. M. van Steenberg. The Design of Focus Area Maturity Models / M. van Steenberg, R. Bos, S. Brinkkemper, et al. – 2010.
24. J. Becker. Developing Maturity Models for IT Management – A Procedure Model and its Application / J. Becker, R. Knackstedt, J. Pöppelbuß // Business & Information Systems Engineering. – 2009. – N 3.
25. A.M. Maier. Developing maturity grids for assessing organisational capabilities: Practitioner guidance / A.M. Maier, J. Moultrie, P.J. Clarkson. – 2009.
26. C.P.Halvorsen. A Taxonomy to Compare SPI Frameworks / C.P. Halvorsen, R. Conradi. – 2002.
27. P. Martins. A comparative study of SPI Approaches with ProPAM / P.V. Martins // . – 2006.
28. S. Komi-Sirviö. Development and Evaluation of Software Process Improvement Methods, PhD Diss. / S. Komi-Sirviö. – VTT Electronics. – 2004. – 190 p.
29. T. de Bruin. Understanding the Main Phases of Developing a Maturity Assessment Model / T.d. Bruin, R. Freeze, U. Kulkarni, M. Rosemann // ACIS 2005 Proceedings. – 2005.
30. W. Bekkers. SPM maturity matrix, Technical Report UUCS-2010-013 / W. Bekkers, I.v.d. Weerd. – Utrecht University. – 2010.
31. Brinkkemper S. Process Improvement in Requirements Management: A Method Engineering Approach / S. Brinkkemper, I. van de Weerd, M. Saek // REFSQ'08. – LNCS, Vol. 5025. – 2008. – P. 6-22.
32. Goeken M. IT Governance Frameworks as Methods / M. Goeken, S. Alter // Proc. ICEIS 2008. –, Vol. 32. – 2008. – P. 331-338.
33. van de Weerd I. A product software knowledge infrastructure for situational capability maturation: Vision and case studies in product management / I. van de Weerd, J. Versendaal, S. Brinkkemper // Proc. REFSQ'06. – 2006. – P. 97–112.
34. T. Mettler. Situational Maturity Models as Instrumental Artifacts for Organizational Design / T. Mettler, P. Rohner // Proc. DESRIST'09. – ACM Press. – 2009.
35. S.-J. Huang. Selection priority of process areas based on CMMI continuous representation / S.-J. Huang, W.-M. Han // Information & Management –2006. – Vol. 43 –P. 297–307.