

## НЕКОТОРЫЕ АСПЕКТЫ РАЗРАБОТКИ И ДАЛЬНЕЙШЕГО РАЗВИТИЯ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ УПРАВЛЕНИЯ ТРЕБОВАНИЯМИ ПРИ СОЗДАНИИ СЛОЖНЫХ ПРОГРАММНЫХ СИСТЕМ

Приведены результаты анализа существующих инструментальных средств управления требованиями для программного обеспечения, дается оценка преимуществ их практического применения. Рассмотрены и обобщены недостатки, затрудняющие перспективу их дальнейшего развития. Предложено одно из направлений их дальнейшего совершенствования и развития.

### *Введение*

Прогресс современного развития компьютерной техники и программного обеспечения предоставил возможность в настоящее время создавать сложнейшие программно-технические комплексы, которые позволяют автоматизировать процессы в различных отраслях жизнедеятельности человека. При этом качественно изменился и сам процесс программирования. Новые средства написания программ, такие как объектно-ориентированное программирование, различные типы систем управления базами данных, язык универсального моделирования UML и т.д., позволяют за короткий срок относительно небольшим коллективам программистов выполнять разработку программного обеспечения (ПО) для решения сложнейших задач. Исследования показали, что все существующее в настоящее время разработки ПО условно можно разбить на три следующие категории [1].

■ Информационные системы и другие приложения, разработанные для использования внутри компании. Эта категория является основой индустрии информационных систем/информационных технологий, или IS/IT.

■ Программное обеспечение, разрабатываемое и продаваемое как коммерческий продукт. Разрабатывающие такое ПО компании обычно называют независимыми поставщиками программного обеспечения, или ISV.

■ Программное обеспечение компьютеров, встраиваемое в другие приборы, машины или сложные системы.

Программное обеспечение такого типа назовем приложениями для встроенных систем или встроенными приложениями.

Эти три типа приложений существенно отличаются по своей природе и имеют различные стратегические цели. Однако все они разрабатываются для выполнения конкретных задач и должны удовлетворять определенному набору требований. В связи с необходимостью планирования и систематизации выполнения работ по проектированию, разработке и внедрению таких программных продуктов для решения сложных заданий, а главное, чтобы функциональные и прочие характеристики конечного продукта соответствовали требованиям заказчика, возникла необходимость создания и развития направления по разработке и управлению требованиями для них. Большинство существующих методов и инструментальных средств, используемых для разработки и управления требованиями ПО, применимы для всех типов приложений.

Данное направление состоит из двух основных процессов: разработка требований для программного обеспечения и управление ими. Процесс разработки требований в настоящее время достаточно изучен, имеет множество классических методов и средств получения, анализа и их классификации. Стадия разработки подразумевает сбор информации, анализ, уточнение и утверждение требований. Как правило, она заканчивается созданием пакета документов: об образе и границах продукта, документации к вариантам использова-

ния, спецификации требований к ПО, словаря данных и соответствующих моделей анализа. После проверки и одобрения этот пакет определяет базовую версию требований, соглашение между разработчиками и клиентами. Такой процесс является относительно статичным и бывает актуальным в основном на начальном этапе проектирования ПО. Иная ситуация складывается при организации процесса управления требованиями. Этот процесс является динамичным и остается актуальным на протяжении всего жизненного цикла ПО. Соглашение по требованиям является связующим звеном между разработкой и управлением требованиями. Дадим определение этому процессу.

*Процесс управления требованиями (ПУТ)* состоит в выявлении, организации, документировании и изменении требований к ПО, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и разработчиками этого ПО.

В силу всего вышесказанного анализ существующих методов и инструментальных средств (ИС) автоматизации процессов управления требованиями, поиск и оценка перспективных направлений дальнейших разработок в этой области в настоящее время является актуальной проблемой.

### **1. Анализ существующих инструментальных средств управления требованиями при разработке ПО**

Под процессом управления требованиями подразумевают все действия, поддерживающие целостность, точность и своевременность обновления соглашения о требованиях в ходе всего жизненного цикла ПО. Основными составляющими ПУТ являются [2]:

- управление изменениями базовой версии требований;
- поддержка планов проекта актуальными в соответствии с меняющимися требованиями;
- управление версиями отдельных требований и документации требований и контроль их состояния;

- контроль состояния требований в базовой версии;
- управление логическими связями между отдельными требованиями, другими материалами для работы с проектом и контроль их корректности.

Обычный документальный способ формирования и хранения требований имеет ряд недостатков, которые определяются необходимостью:

- обновления и синхронизации документов;
- хранения дополнительной информации атрибутов для каждого требования;
- определения взаимосвязей между функциональными требованиями и другими элементами системы;
- осуществления ручной передачи изменений всеми членами группы разработчиков ПО;
- трассирования статуса требований, который представляет собой трудоемкий и неудобный процесс;
- одновременного управление наборами требований, запланированных для различных выпусков продуктов;
- дублирования требований для различных версий;
- модификации требований несколькими территориально распределенным коллективам разработчиков ПО;
- дополнительного сохранения отклоненных или удаленных из основной версии требований и т.д.

ИС управления требованиями, хранящие информацию в многопользовательской базе данных, позволяют устранить эти недостатки. В небольших проектах для управления требованиями иногда достаточно использовать электронные таблицы или простые базы данных, сохраняя текст и отдельные атрибуты каждого требования. В более крупных проектах выгодно применять коммерческие средства управления требованиями, которые позволяют пользователям: импортировать требования из исходных документов; определять зна-

чения атрибутов; фильтровать и выводить на экран содержание базы данных; экспортировать требования в различных форматах; контролировать связи трассирования и соединять требования с элементами, хранящимися в других средствах разработки ПО.

Таких средств в настоящее время довольно много, они продолжают развиваться и их возможности расширяются с каждой версией. Проведенный анализ всего этого многообразия показал, что наиболее распространены несколько разработок. Данные о производителях этих инструментальных средств управления требованиями и их основных функциональных свойствах приведены в таблице [3, 4]. Важное отличие инструментальных средств заключается в способе хранения данных.

Большинство из них хранит все требования, атрибуты и информацию трассирования в базе данных. В зависимости от продукта база данных может быть коммерческой или разработанной собственными силами и запатентованной, реляционной или объектно-ориентированной. Требования могут импортироваться из различных исходных документов, но затем они сохраняются в базе данных. Как правило, текстовое описание требования считается просто

одним из атрибутов. Некоторые продукты позволяют устанавливать связи отдельных требований с внешними файлами (например, файлами Microsoft Word, Microsoft Excel, графическими файлами и т.д.), в которых содержится информация, дополняющая содержимое базы.

При документальном подходе документ, созданный при помощи текстового процессора (такого, как Microsoft Word или Adobe FrameMaker), считается основным хранилищем требований. RequisitePro позволяет выбирать текстовые строки в документе Word для сохранения их в виде отдельных требований в базе данных. После ввода требований в базу данных можно определить атрибуты и связи трассирования так же, как и в продуктах, данные которых хранятся в БД. Механизмы синхронизации базы данных и содержимого документа для этого предусмотрены. RTM Workshop поддерживает обе схемы: в первую очередь хранение данных в базе данных, но также и в документе Microsoft Word.

**1.1. Преимущества использования инструментальных средств управления требованиями.** Даже если была выполнена большая работа по сбору требований для своего проекта, автоматизированная поддержка поможет

Таблица

Инструмент	Производитель	Способ хранения атрибутов в базе данных	Связь требований с внешними ИС
Active! Focus	Harware Technologies	+	-
CaliberRM	Borland Software Corporation	+	+
C.A.R.E.	SOPHIST Group	+	-
DOORS	Telelogic	+	+
RequisitePro	Rational Software Corporation	-	+
RMTrak	RBC, Inc.	-	-
RTM Workshop	Integrated Chipware, Inc.	+	+
Slate	EDS	+	+
Vital Link	Compliance Automation, Inc.	-	+

оперировать этими требованиями в процессе разработки. Инструментальное средство управления требованиями становится полезным со временем, когда детали требований постепенно забываются разработчиками ПО. Опишем некоторые задачи, которые подобное средство поможет решать.

**Управление версиями и изменениями.** Программный проект должен определять четкий набор требований для конкретной версии программного обеспечения. Некоторые средства управления требованиями предлагают функции гибкого управления базовой версией. Они также сохраняют историю изменений каждого требования. Предоставляется возможность записывать обоснование каждого решения об изменении и при необходимости вернуться к предыдущей версии требования. Некоторые средства, такие, как Active! Focus и DOORS, содержат простые встроенные системы изменений-предложений, устанавливающие связи между предложениями об изменениях и измененными требованиями.

**Хранение атрибутов требований.** Для каждого требования необходимо записывать несколько описательных атрибутов. Каждый, кто работает над проектом, должен иметь доступ к просмотру этих атрибутов, а некоторые к изменению их значений. Инструментальное средство управления требованиями генерирует несколько системных атрибутов, например дату создания требования и номер его версии, а также позволяет создавать дополнительные атрибуты различных типов данных. Продуманное определение атрибутов даёт возможность всем заинтересованным в проекте лицам просматривать подмножества требований, основанных на выбранных комбинациях значений атрибутов. Можно запросить список всех требований, основанных на каком-либо бизнес-правиле, чтобы принять решение о последствиях его изменения. Один из способов учета требований в основных версиях различных выпусков продукта – использовать атрибут «номер выпуска».

**Облегчение анализа воздействия.** Средства управления требованиями помогают осуществлять трассирование требований, позволяя определять связи между различными типами требований, между требованиями в различных подсистемах и между отдельными требованиями и связанными системными компонентами (например, дизайном, модулями кода, тестами и пользовательской документацией). Эти связи помогают анализировать воздействие, которое предлагаемое изменение окажет на конкретное требование, выявляя другие элементы системы, которые оно затронет. Другое полезное свойство возможность отследить историю каждого функционального требования в обратную сторону, вплоть до первоисточника.

**Трассирование статусов требований.** Собрав требования в базе данных, можно узнать, сколько отдельных требований было указано для продукта. Трассирование статуса каждого требования в процессе разработки способствует общему трассированию статуса проекта. Менеджер проекта точно представляет себе статус проекта, если знает, какая часть требований для следующего выпуска уже проверена, какая реализована, но не проверена, а какая часть еще не до конца реализована.

**Контролируемый доступ.** Средства управления требованиями позволяют устанавливать права доступа для отдельных пользователей и их групп, предоставлять информацию для общего доступа через Web-интерфейс к базе данных территориально распределенному коллективу разработчиков ПО. Базы данных используют блокировку на уровне требований, позволяя многим пользователям обновлять содержимое базы данных одновременно.

**Связь со всеми заинтересованными в проекте лицами.** Некоторые средства управления требованиями позволяют членам команды, которая создана для участия в проектировании и разработке ПО, обсуждать вопросы, связанные с требованиями, на тематических дискуссиях с помощью электронных

средств обмена сообщениями. Автоматически отсылаемые электронные сообщения уведомляют членов команды, когда в дискуссии появляется новая запись или какое-либо требование модифицируется. Возможность работы с требованиями через Интернет сокращает расходы на транспорт и уменьшает документооборот.

**Повторное использование требований.** Хранение требований в базе данных облегчает их повторное использование в других проектах и подпроектах. Требования, логически подходящие нескольким разделам описания проекта, можно сохранить однажды, а затем лишь ссылаться на них во избежание дублирования.

**1.2. Возможности инструментальных средств управления требованиями.** Все существующие разработки инструментальных средств управления требованиями имеют одну общую цель - дать разработчикам возможность автоматизировать процессы оперативного управления этими требованиями на протяжении всего жизненного цикла разработки, внедрения и эксплуатации ПО. Однако практическая реализация этих средств, их функциональные свойства, наличие связи и обмена данными с другими инструментариями для разработки ПО осуществляются различными способами. Рассмотрим некоторые из них.

Коммерческие средства управления требованиями позволяют определять различные типы (или классы) требований, такие, как бизнес-требования, варианты использования, функциональные требования, требования к оборудованию, а также ограничения. Это позволяет отделять задачи, с которыми можно работать как с требованиями, от другой полезной информации, содержащейся в спецификации требований к ПО. Все инструментальные средства обладают мощными возможностями определения атрибутов для каждого типа требований, что является большим преимуществом перед обычными способами документирования спецификации требований к ПО.

Большинство инструментальных средств управления требованиями в той или иной степени интегрируются с Microsoft Word, на панель меню Microsoft Word добавляется их специализированный раздел. Vital Link основан на Adobe FrameMaker, а Slate интегрируется и с FrameMaker и с Word. Инструментальные средства высокого уровня поддерживают широкий набор форматов файлов для импорта и экспорта. Некоторые инструменты позволяют пометить текст в документе Word, чтобы обращаться с ним как с отдельным требованием. Инструмент выделяет требование и вставляет в документ закладки Word и скрытый текст. Кроме того, пользователь получает возможность различными способами анализировать документы, чтобы извлекать из них отдельные требования. Анализ документа в текстовом редакторе будет несовершенным, если только при создании документа пользователь не станет последовательно использовать стили или ключевые слова, такие, как «должно».

Инструментальные средства поддерживают иерархическую систему нумерации требований помимо уникального внутреннего идентификатора для каждого требования. Эти идентификаторы обычно состоят из короткого текстового префикса, который указывает на тип требования, например UR обозначает требование пользователя (user requirement), и уникального целого числа. Некоторые средства предоставляют возможность пользоваться Microsoft Windows Explorer для отображения и управления иерархическим деревом требований. Один из способов отображения требований в DOORS - иерархически структурированная спецификация требований к ПО.

К возможностям вывода данных инструментальных средств относится способность генерировать требования либо в документе заданного пользователем формата, либо в табличном отчете. CaliberRM обладает мощной функцией Document Factory, позволяющей определить шаблон спецификации требований

к ПО в Word, используя простые способы для разметки макета страницы, шаблона текстов, атрибутов для извлечения данных из базы и стилей текста, которые пользователь хочет применять. Document Factory заполняет этот шаблон информацией, которую выбирает из базы данных соответственно критериям заданного пользователем запроса, чтобы создать заказанный документ спецификации требований к ПО. Таким образом, спецификация требований к ПО, в сущности, представляет собой отчет, генерируемый по некоторой выборке из содержимого базы данных.

Все инструментальные средства обладают развитыми возможностями трассирования. Например, в RTM Workshop каждый проект определяет схему классов, похожую на диаграмму „сущность – связь” для всех хранящихся типов объектов. Трассирование выполняют, определяя связи между объектами двух классов (или внутри одного класса) на основе взаимоотношений между классами, заданными в схеме.

К числу других функций относится возможность задавать группы пользователей и определять разрешения некоторым пользователям или группам на создание, чтение, обновление и удаление проектов требований, атрибутов и их значений. Некоторые продукты позволяют поместить нетекстовые объ-

екты, такие, как графику и крупноформатные таблицы, в базу требований. Инструментальные средства предлагают также средства обучения или проекты-примеры, которые помогут пользователям научиться работать быстро и эффективно.

Эти продукты отражают тенденцию к увеличению интеграции с другими инструментальными средствами, используемыми в разработке приложений, как показано на рисунке. При выборе средства для управления требованиями необходимо выяснить, сможет ли оно обмениваться данными с другими используемыми пользователем инструментами. Приведем примеры взаимосвязей между инструментальными средствами существующими на сегодня:

- пользователь может устанавливать связи между требованиями в RequisitePro и вариантами использования, смоделированными в Rational Rose, а также вариантами тестирования, хранящимися в Rational TeamTest;

- DOORS позволяет трассировать требования вплоть до отдельных конструктивных элементов, хранящихся в Rational Rose, Telelogic Tau и других инструментах дизайна;

- RequisitePro и DOORS могут устанавливать связи между отдельными элементами проектного задания в Microsoft Project;



Рисунок. Интеграция инструментальных средств управления требованиями с другими видами программных средств

- CaliberRM имеет централизованную структуру коммуникаций, которая позволяет пользователю связывать требования с вариантами использования, классами или элементами дизайна процессов, хранящимися в TogetherSoft Control Center, с исходным кодом, хранящемся в Borland StarTeam, и с тестовыми элементами, хранящимися в Mercury Interactive's TestDirector. Он сможет получать доступ к этим взаимосвязанным элементам непосредственно из требований, хранящихся в базе данных CaliberRM;

- RequisitePro позволяет взаимодействовать с базами данных, созданных для Microsoft Access, Microsoft SQL Server и Oracle.

Оценивая такие программные средства, необходимо учесть преимущества интеграции продуктов при составлении требований, тестировании, трассировании проекта и других процессов. Например, как опубликовать основной набор требований в инструменте для управления версиями продукта и определить связи трассирования между функциональными требованиями и конкретными элементами дизайна или кода.

### ***2. Особенности и перспективы дальнейшего развития инструментальных средств управления требованиями***

Для эффективной организации и управления процессами проектирования и разработки сложных программных систем в настоящее время требуется использование различных инструментальных средств, которые позволяют проектировщикам и разработчикам ПО автоматизировать выполнение этих процессов. Одним из основных является инструментальное средство управления требованиями. Любое такое инструментальное средство позволяет создать структурную схему для организации процесса управления требованиями, значительно сократить время выполнения отдельных операций этого процесса, что в итоге дает возможность сократить время проектирования и разработки самого ПО, учесть все корректные и не-

противоречивые требования заказчика. Как и любое другое инструментальное средство аналогичного типа, данный инструментальный требует определенной подготовки и необходимой квалификации пользователей при работе с ним. Существенное преимущество от использования такого инструментального средства можно получить при проектировании и разработке ПО для больших и сложных задач, в разработке которых принимают участие многочисленные коллективы разработчиков (от 10 человек и более). Различные функциональные и прочие свойства существующих коммерческих инструментальных средств требуют от разработчиков на начальном этапе проектирования ПО проведения анализа этих средств, чтобы выбрать такое, которое лучше других поможет успешно выполнить разработку требуемого ПО. При выборе инструментального средства следует учитывать такие параметры, как платформа, цена, режим доступа и способ хранения данных для требований – документальный или в базе данных, которые лучше всего подходят данной среде разработки и культуре. Инструментальное средство открывает доступ к требованиям в базе данных любому разработчику ПО, имеющему соответствующие разрешения.

Однако, несмотря на вышеописанные и некоторые другие объективные организационные преимущества работы в такой среде, все существующие на данный момент коммерческие инструментальные средства управления требованиями обладают некоторыми общими внутренними конструктивными недостатками. Наиболее существенные среди них зачастую оказывают решающее значение на окончательный выбор при покупке такого инструментального средства. Перечислим главные из них:

- сложность настройки и эксплуатации таких инструментальных средств пользователем;

- относительно высокая их стоимость;

- отсутствие или ограниченность связей автоматического обмена информацией с другими инструментальными средствами проектирования, разработки и отладки ПО.

Необходимость поиска возможных направлений для преодоления этих недостатков разработки существующих инструментальных средств, которые на первый взгляд являются взаимоисключающими, потребовала проведения более углубленных исследований в этом направлении. Результаты анализа существующей проблемы позволили наметить несколько возможных путей для ее преодоления. Одним из наиболее оптимальных, на наш взгляд, является решение, которое предложено и описано в данной статье. Оно не заменит тех необходимых организационных потребностей, которые необходимы при работе с таким инструментальным средством, однако позволяет устранить конструктивные проблемы и недостатки самого инструментального средства, расширить его функциональные возможности и существенно уменьшить цену продажи конечным пользователям. В обобщенном виде структура и перечень необходимых составляющих предлагаемого варианта для разработки такого инструментального средства управления требованиями должны соответствовать следующим обязательным условиям:

- базовым средством хранения информации о требованиях должна быть многопользовательская база данных;

- организация формирования всех требований в инструментальном средстве должна осуществляться по иерархическому принципу от потребностей заказчика через функции к программным требованиям к ПО;

- с целью оперативности и упрощения управления требованиями для каждого из них требуется добавить параметры (весовые коэффициенты): приоритетность, трудоемкость реализации, степень риска;

- необходимость наличия взаимосвязей между различными требованиями и структура их организации должны по-

зволять организовывать такие связи как линейно по горизонтали для требований одного уровня, так и иерархически по вертикали для требований разных уровней;

- построение соответствующего однотипного универсального интерфейса пользователя, желательно на различных языках, для работы с различными компонентами самого инструментального средства управления требованиями;

- одной из основных составляющих предлагаемого варианта является необходимость разработки возможности расширенного доступа и взаимообмена информацией этого инструментального средства с большинством, а в идеальном случае со всеми другими инструментальными средствами проектирования и разработки ПО через универсальный интерфейс;

- для удобства и наглядности работы разработчиков с требованиями необходимо создание компонента для возможности генерирования и представления этих требований в произвольном задаваемом пользователем формате.

Организация хранилища информации о требованиях в многопользовательской базе данных является одним из основных условий, так как только в этом случае может быть задействован в полной мере гибкий механизм организации управления требованиями. Любой другой вариант существенно ограничивает такие возможности и, на наш взгляд, неприемлем для полноценного построения инструментария. Большинство существующих средств уже используют базы данных, что подтверждает обязательность такого требования. При этом появляется дополнительная возможность формирования и управления требованиями на разных языках. Это актуально для международного коллектива разработчиков, когда с одними и теми же требованиями работают специалисты, говорящие на разных языках. В сочетании с обменом информацией через Интернет такая функция позволяет, например, формиро-

вать территориально распределенные коллективы, которые участвуют в проектировании и разработке ПО для решения больших и сложных, а в некоторых случаях и глобальных задач.

Иерархическая организация требований по принципу от наиболее общих потребностей через функции к программным требованиям значительно упрощает пользователю процесс формирования и управления требованиями при проектировании и разработке ПО и позволяет детализацию требований проводить поэтапно по мере надобности. Такая структура предоставляет гибкий инструмент формирования и управления требованиями различных уровней для конкретных версий. Проектировщики и разработчики ПО получают возможность на начальном этапе проанализировать и отобрать все или большинство требований-потребностей, предъявляемых для разработки такой системы. Затем с помощью функций проводится предварительная оценка необходимых показателей данного программного проекта (время разработки, стоимость, необходимые ресурсы и т.д.). При согласовании и уточнении всех необходимых формальностей с заказчиком одним из важных пунктов является определение приоритетности и последовательности реализации этих потребностей. Таким образом, разработка и управление непосредственно программных требований осуществляются непосредственно перед началом разработки той версии, в которую данное требование вошло. Это позволяет экономить время не детализируя разработку всех требований на начальном этапе проектирования, а приступать к нему тогда, когда оно запланировано. Однако следует учитывать тот факт, что в процессе разработки требования могут изменяться, добавляться или удаляться. Особенно это актуально для современных моделей жизненного цикла программных систем. Например, для развитой спиральной модели, описанной в [5].

Наличие весовых коэффициентов приоритетности, трудоемкости реализации и степени риска для требований по-

зволяет разработчикам распределять и оптимизировать проектирование и разработку сложных программных систем во времени и последовательности реализации. Существенным преимуществом наличия таких параметров является возможность проведения оперативного проектирования изменений в системе путем корректирования коэффициентов уже в процессе выполнения разработки ПО, если возникает необходимость. Такая возможность позволяет автоматизировать сам процесс управления требованиями на всех этапах разработки ПО и быстро, без существенных затрат, проектировать всю системы с учетом изменившихся обстоятельств. При проектировании инструментального средства управления требованиями весовые коэффициенты могут присутствовать в качестве основных атрибутов для требований.

Система наличия взаимосвязей между требованиями позволяет получить разработчику ПО наглядную картину проектируемой системы или какого-либо ее варианта в целом в любой момент времени, оценить степень зависимости требований между собой и сложность выполнения изменения или модификации того или иного требования еще до начала проведения таких изменений. Такая структура предоставляет инструментальному средству управления требованиями дополнительную возможность моделировать процесс выполнения всех взаимосвязанных изменений при условии необходимости изменения какого-либо требования. Основываясь на результатах различных вариантов такого моделирования, позволяющих оценить объем необходимых затрат для выполнения изменений, по согласованию с заказчиком, можно осуществлять поиск и принимать наиболее оптимальный и приемлемый вариант выполнения внесенных изменений.

Создание однотипного универсального интернационального интерфейса пользователя для работы с различными компонентами внутри самого инструментального средства управления

требованиями является дополнительным, однако весьма важным и необходимым сервисом. Важнейшим преимуществом такого компонента является тот факт, что для пользователя отпадает необходимость в многочасовом или даже многодневном изучении инструкций для работы с такой системой и появляется возможность непосредственно приступить к работе с ней. В общем виде такой алгоритм предусматривает начало работы пользователя с любым компонентом системы из одной общей панели, затем с помощью навигатора и дополнительных помощников-мастеров предоставляется существующий сервис системы и проводится сопровождение его действий на всех этапах нахождения в системе.

Формирование возможности расширенного доступа и обмена информацией с другими инструментальными средствами проектирования и разработки ПО через универсальный интерфейс является одной из важнейших составляющих предлагаемого инструментального средства. Такая возможность позволяет разработчикам сложных программных систем перейти на новый, более высокий уровень выполнения работ - комплексную автоматизацию процессов проектирования, разработки, тестирования и внедрения программных средств. При этом отпадает необходимость дублирования хранения одинаковой информации в различных инструментальных средствах. Достаточно хранить ее в одном источнике, а во всех остальных организовать связь и ссылку на нее. Комплексная автоматизация процессов, с одной стороны, значительно упрощает их проектирование, а в случае необходимости внесения необходимых корректировок, с другой – существенно сокращает время на выполнения этих работ.

Необходимость хранения информации о требованиях в базе данных наряду с перечисленными выше преимуществами обладает одним недостатком – недостаточная наглядность в оперативной работе с ними. С целью устранения

этого недостатка предлагается разработка дополнительного программного модуля в таком инструментальном средстве, который генерирует возможность представления требований в произвольном задаваемом пользователем формате. Этот дополнительный сервис, с одной стороны, позволяет хранить в базе данных требования различных типов в одном стандартном формате, с другой стороны, при необходимости представления их пользователю в визуальном или печатном виде осуществляется обратное преобразование в соответствующий формат. Следует заметить, что сам первичный процесс формирования таких разнотипных требований осуществляется с помощью этого же дополнительного сервисного модуля.

Простота в работе, универсальность и «многоязычность» пользовательского интерфейса и главное наличие связи и обмена данными с другими различными инструментальными средствами проектирования и разработки ПО, благодаря дополнительно предложенным составляющим, должны заинтересовать и привлечь значительно большую аудиторию потенциальных пользователей и потребителей данного продукта в сравнении с существующими аналогами. Этот фактор, несмотря на большие затраты при проектировании и разработке такого инструментального средства, позволит снизить его цену продажи до уровня, который позволит приобретать продукт средним и даже мелким предприятиям и разработчикам ПО. Такой дополнительный класс потенциальных пользователей и заказчиков предлагаемого инструментального средства в свою очередь будет способствовать еще большему расширению рынка потребления.

### *Заключение*

Основной целью предложенных в данной работе структурной особенности и перечня необходимых составляющих компонентов нового инструментального средства управления требованиями явилась необходимость поиска возможных

направлений для преодоления конструктивных недостатков существующих разработок в этой области. Исследования показали эффективность применения данного средства. Предложен комплексный вариант разработки, который позволяет наряду с существенным расширением функциональных возможностей значительно снизить ее цену продажи.

Следует особенно отметить актуальность разработки и использования этого инструментального средства при переходе к перспективному направлению комплексной автоматизации всего процесса проектирования, разработки и отладки сложных программных систем с помощью нескольких различных ИС. Наличие интерфейса связи и взаимообмена информацией между данным средством и другими ИС, возможность автоматизации процесса анализа необходимости внесения изменений и корректировок при внесении изменений в требования, формирование перечня таких изменений и оперативное занесение их в другие ИС позволяют утверждать, что ведущая роль в таком комплексе может принадлежать инструментальному средству управления требованиями к ПО с учетом описанных в данной статье рекомендаций и предложений к нему.

1. *Леффингуел Д, Уидрих Д.* Принципы работы с требованиями к программному обеспечению. Унифицированный подход: Пер. с англ. – М.: Изд. дом «Вильямс», 2002. – 448 с.
2. *Вигерс К.* Разработка требований к программному обеспечению: Пер. с англ. – М.: Изд. дом «Русская Редакция», 2004. – 576 с.
3. *Jones. D. A., Donald M. Y., John F. N., J.Simpson.* Factors Influencing Requirement Management Toolset Selection. Proc. of the Fifth Annual Symp. of the National Council on

Systems Engineering., Seattle, 1995. – 2. – p.33–58.

4. Лесин Д. А., Томащенко С. Н.. Основы использования Rational RequisitePro. – 2001, 214 с.

5. *Алексеев В.А., Терещенко В.С.* Развитие спиральной модели жизненного цикла программных систем // Проблемы программирования. – 2003. – № 4. – С. 34-42.

*Получено 22.02.05*

### **Об авторах**

*Лаба Михаил Николаевич*

заведующий отделом

*Матвейшин Сергей Николаевич*

кандидат технических наук,

научный сотрудник

### *Место работы авторов*

Институт программных систем

НАН Украины,

просп. Академика Глушкова, 40,

Киев-187, 03680, Украина

Тел. 526 3225, 526 7038

E-mail: sergmmm@isofts.kiev.ua