

Розв'яжемо рівняння  $4 * \sin(x) + 3 * \cos(x) = 1$  на проміжку  $0 \leq x \leq 9$ . Як видно з графіка, на цьому діапазоні існує три корені. Результат роботи програми в даному випадку наведено на рисунках 7, 8.

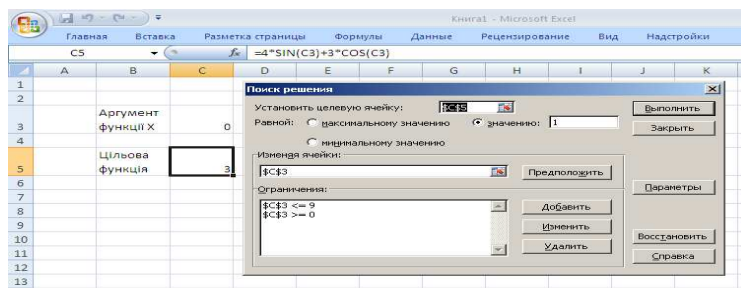


Рис. 7. Вихідні дані для розв'язку рівняння  $4 * \sin(x) + 3 * \cos(x) = 1$

Звернемо увагу на те, що початкове значення аргументу спочатку не задавалось. Отриманий корінь рівняння є вірним, але він є одним з трьох коренів (рис. 6). При цьому даний корінь є середнім коренем із трьох можливих. Не приводячи результатів обчислень зазначимо, що при присвоєнні початковому значенню аргументу значення 0,1, тобто при  $x = 0,1$  відповідного рішення не знайдено, а при значенні аргументу  $x = 1$  знайдено перший від початку системи координат корінь  $x = 2,4$ .

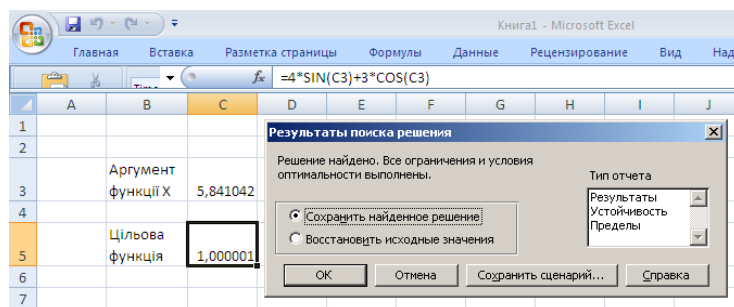


Рис. 8. Результати розв'язку рівняння  $4 * \sin(x) + 3 * \cos(x) = 1$

### Висновки

Отже, в результаті проведених досліджень, можна сказати, що надбудова табличного процесора MS Excel «Пошук рішення» є ефективним засобом розв'язування нелінійних рівнянь. В більшості випадків, при наявності одного кореня програма дає дійсні результати. Однак, в деяких випадках результати є помилковими.

При некоректному використанні програми, коли в заданому діапазоні не існує коренів або якщо їх кількість більше одного, необхідно передбачати якісно результат і здійснювати аналіз та перевірку отриманих рішень.

**Список літератури:** 1. Должанков В., Колесніков Ю. Microsoft Excel 2002. – СПб: БХВ – Петербург, 2003.-1072 с. 2. www. Msexcel.ru Профессиональные приемы работы в Microsoft Excel 3.WWW. Office. Microsoft.com/excel – help/HP010072691.aspx.

*Поступила в редколлегию 20.06.2012*

**УДК 004.9, 654.01, 658.512, 69.059.7, 519.6**

**А.В.КАЛМЫКОВ**, канд.техн.наук, НАУ им. Н.Е. Жуковского «ХАИ», Харьков

## МЕТОДОЛОГИИ RUP И AGILE В УПРАВЛЕНИИ ИТ-ПРОЕКТАМИ ТЕЛЕКОММУНИКАЦИОННОГО ПРЕДПРИЯТИЯ

У статті розглядається використання протягом життєвого циклу розробки і супроводу ІТ-систем телекомунікаційного підприємства різних методологій створення програмного забезпечення. Показано можливості їх ефективного застосування та задоволення суперечливих

вимог на основі декомпозиції функціональної архітектури системи.

**Ключові слова:** RUP, Agile, управління проектами, життєвий цикл.

В статье рассматривается использование в течение жизненного цикла разработки и сопровождения IT-систем телекоммуникационного предприятия различных методологий создания программного обеспечения. Показаны возможности их эффективного применения и удовлетворения противоречивых требований на основе декомпозиции функциональной архитектуры системы.

**Ключевые слова:** RUP, Agile, управление проектами, жизненный цикл

The article considers the usage of different methodologies of software development during a life-cycle of IT-systems (including designing and further maintenance), which relate to the telecommunication industry. It is shown the possibility of their effective application and compliance to contradictory demands on the basis of the functional architecture decomposition.

**Keywords:** RUP, Agile, project management, life cycle

## **Введение**

Жизненный цикл производственных корпоративных IT-систем состоит из нескольких последовательных этапов проектирования, разработки, внедрения, эксплуатации [1]. При этом, как правило, фактическое интенсивное развитие информационных систем не прекращается с завершением этапа внедрения. Например, системы, обеспечивающие деятельность операторов телекоммуникаций, непрерывно развиваются в течение всего периода эксплуатации в виду необходимости внедрения и предоставления абонентам новых видов услуг, расширения существующих сервисов, наращивания нагрузок и объемов обрабатываемой информации.

Процессы внедрения, разработки и сопровождения информационных систем, обеспечивающих предоставление телекоммуникационных услуг, характеризуются четкими временными рамками начала и завершения работ, ресурсами, выделенными для их выполнения, и ожидаемыми результатами. Данные активности имеют все признаки проектной деятельности, и, следовательно, управление ими целесообразно осуществлять на основе принципов проектного менеджмента.

Следует отметить, что с одной стороны, необходимо обеспечить высокое качество реализуемых в телекоммуникационной отрасли IT-решений, преемственность их развития, что достигается использованием традиционных фундаментальных подходов в управлении подобными разработками. С другой стороны, высокая динамика изменений требований, характеристик и видов предоставляемых телекоммуникационных услуг вызывает необходимость применения в разработке и внедрении компонентов IT-систем (OSS/BSS), реализующих такие услуги, гибких, адаптивных методологий управления соответствующими проектами и программами.

## **Известные подходы к управлению IT-проектами**

Традиционные подходы в управлении проектами разработки информационных предполагают создание различных сопровождающих артефактов, каких как: устав проекта, план проекта, план коммуникаций [2]. Как правило, разработка данных сущностей выполняется на начальной стадии проекта и требует привлечения достаточно квалифицированных исполнителей.

Кроме того, предусматривается создание различной сопровождающей и сдаточной документации: описание решения, спецификация ПО, различные руководства и справочники по эксплуатации. Например, такие действия предполагаются в методологии RUP [3], успешно применяемой для управления разработкой больших информационных систем. На рис. 1 показаны последовательность этапов разработки программной продукции в соответствии с данной методологией, ожидаемое распределение нагрузки и потребности в ресурсах в течение выполнения проекта.

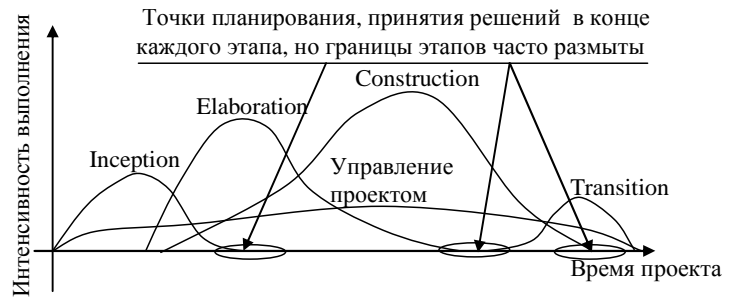


Рис. 1. Методология разработки RUP, точки принятия решений и этапы

Вместе с тем, такой подход является чрезмерно затратным для небольших изменений в конфигурациях системы или при развитии уже существующей функциональности. Такие работы, как правило, имеют небольшие длительность и трудоемкости, сравнимые с подготовкой и сопровождением артефактов, требующихся в соответствии с методологией RUP. Кроме того, подобные активности часто неоднократно повторяются в течение определенного календарного периода времени. Поэтому такой характер процесса разработки хорошо согласуется с методологиями семейства Agile [4, 5] (см. рис. 2), согласно которым управление проектами разработки осуществляется посредством разбиения проекта на последовательность относительно независимых коротких итераций (sprint). Каждая такая итерация содержит все активности, характерные для «большого» проекта.

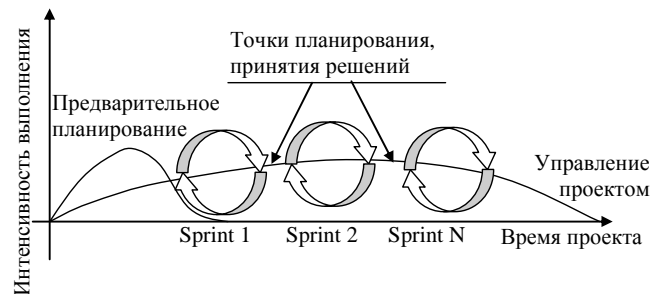


Рис. 2. Методология разработки Agile, итерации и точки принятия решений

В табл. 1 показаны ключевые отличия данных методологий организации и управления

проектами разработки, внедрения и сопровождения информационных систем [6].

Методологии RUP и семейства Agile можно рассматривать как два противоположных взгляда на разработку программного обеспечения, поэтому особенности управления IT-проектами имеет смысл рассматривать с позиций этих двух подходов.

Управление разработкой сложных информационных систем во многих случаях организовывается с учетом особенностей разрабатываемого объекта. Например, в работе [7] показана декомпозиция IT-системы телекоммуникационного предприятия на несколько функциональных уровней, относительно которых организовывается управление разработкой (см. рис. 3): адаптация приложений, специализированные процессы, общие сервисы и прикладное ПО, программно-аппаратные комплексы.

Таблица 1. Ключевые особенности методологий RUP и Agile-подобных.

<b>RUP</b>	<b>Agile-подобные(на примере SCRUM)</b>
Итерационный подход. Проект состоит из 4-х этапов, которые частично могут выполняться одновременно и параллельно	Итерационный подход. Каждый Sprint (итерация) является относительно независимой частью проекта
Детальный план проекта с четким окончанием и промежуточными этапами (milestone)	Высокоуровневый план проекта, планирование каждой итерации определяется в конце предыдущей итерации на основе результатов ее выполнения
Объем проекта определяется как можно заранее (перед началом проекта или на этапе вхождения).	В конце каждой итерации оценивается соответствие текущего состояния желаемому (списку запланированных итераций) и определяются объемы для следующей итерации

При этом к традиционной разработке программных решений можно отнести только действия на уровнях специализированных процессов, общих сервисов и прикладного ПО. Действия на уровне адаптации приложений направлены на приспособление (кастомизацию) существующих решений под потребности пользователя, поэтому для их реализации целесообразно применять гибкие подходы, не предусматривающие длительных подготовительных процедур.

Уровень программно-аппаратных платформ предполагает создание необходимой функциональности на основе комплексирования из готовых компонентов. Очевидно, что для управления данными проектными действиями следует использовать разные подходы.

Фактически, в представлении IT-системы, показанном на рис. 3, каждый из функциональных уровней состоит из одной или нескольких взаимодействующих подсистем, каждая

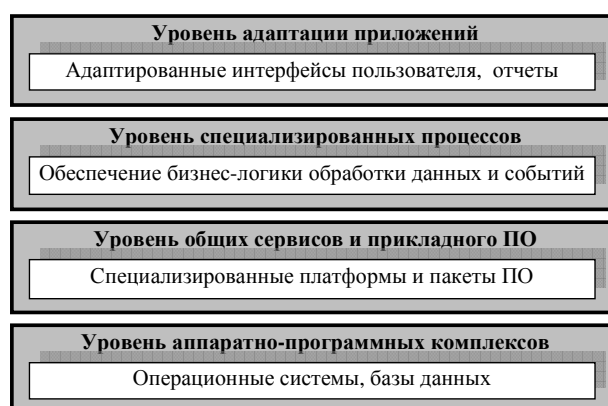


Рис.3. Декомпозиция IT-системы на функциональные уровни

из которых является относительно независимой, и при условии некоторой адаптацией может использоваться в составе других разрабатываемых или действующих информационных систем. Вместе с тем, предложенная высокоуровневая модель, несмотря на предоставляемые возможности по структуризации и декомпозиции состава выполняемых работ и проектных действий, не содержит конкретных рекомендаций по управлению IT-проектами, и, соответственно, её целесообразно применять совместно с известными практическими методиками управления разработкой IT-систем.

В то же время, перечисленные выше методологии управления созданием программной продукции являются слишком абстрактными и не учитывают многие отраслевые особенности разработки и эксплуатации проектируемых IT-систем.

### **Постановка задачи**

Очевидно, что существует некоторый разрыв между существующими

высокоуровневыми подходами к разработке, внедрению информационных систем и практически, детально проработанными концепциями разработки программного обеспечения, что затрудняет их совместное практическое применение в реальных IT-проектах. В частности, это также относится к широко применяемым в настоящее время методологиям RUP, Agile.

Таким образом, представляется целесообразным формирование методики комплексного использования в IT-проектах элементов различных методологий и подходов к управлению разработкой информационных систем разного объема и сложности, основываясь на их преимуществах и недостатках, а также на особенностях объектов проектирования.

### **Декомпозиция процессов разработки на основе анализа архитектуры и особенностей жизненного цикла IT-систем**

В работе [8] предлагается выполнять проектирование сложных информационно-компьютерных систем последовательно по стратам, соответствующим группам характеристик и свойств: цели, функции, структура, информационные потоки, структуры данных, алгоритмы, программное и аппаратное обеспечение 3-х сторон. Такая декомпозиция процесса разработки в целом согласуется с применяемым рядом разработчиков разделением системы на функциональные уровни (см. Рис. 3). В данном случае при разработке сложных

IT-систем, под проектными действиями понимается не только разработка отдельных компонентов системы, но и решение системотехнических задач адаптации, развития и комплексирования существующих подсистем. Разработкам на описанных выше функциональных уровнях сложной информационной системы ставится в соответствие последовательность проектных действий по стратам, как показано на рис. 4.

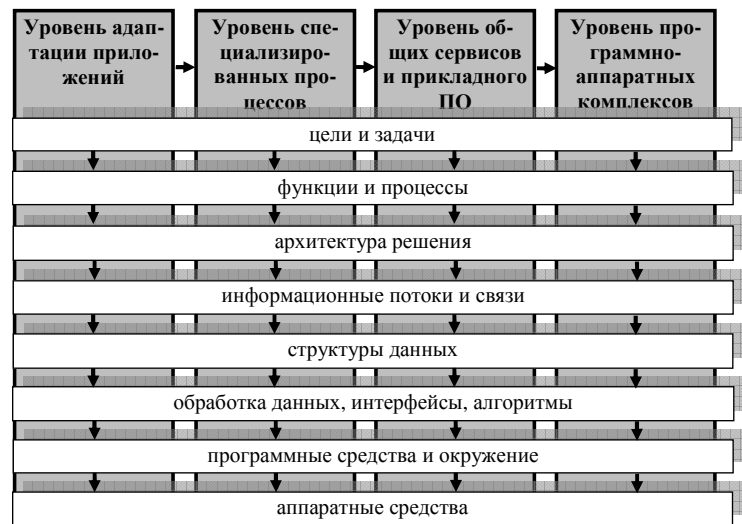


Рис. 4. Стратифицированный подход к разработке IT-систем

При этом проектные действия выполняются последовательно для каждого функционального уровня и по каждой страте представления системы. Очевидно, что длительность, интенсивность и специфика проектных действий на каждом из функциональных уровней зависит от текущего этапа разработки этапа и объемов планируемых изменений в системе, поэтому при выборе методологий управления проектами разработки сложных систем необходимо также учитывать особенности жизненного цикла IT-систем.

Для телекоммуникационной отрасли организацией TMF разработана подробная методология разработки и внедрения информационных систем (OSS/BSS) New Generation Operations Systems and Software (NGOSS) [9], в составе которой сформулирована концепция жизненного цикла информационных систем.

Важнейшими достоинствами этой методологии является учет возможных изменений требований заказчика при эксплуатации системы, описание развития системы как циклично повторяющихся процессов анализа, разработки и внедрения, наличие четкой классификации, иерархии производственных целей, функций телекоммуникационного предприятия и связанных с ними информационных систем. На рис. 5 показано общее представление жизненного цикла разработки систем в нотации TMF NGOSS.

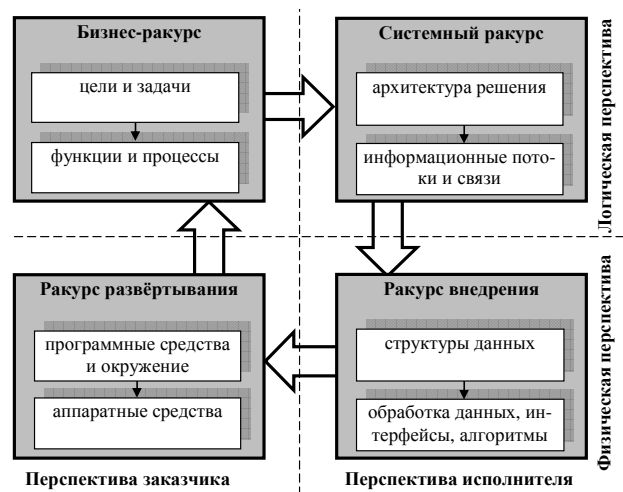


Рис.5. Жизненный цикл IT-систем в соответствии с TMF NGOSS

Из рис. 5 следует, что ракурсы, соответствующие определенным этапам жизненного цикла, отражают позицию заинтересованных сторон проекта (заказчика, исполнителя) и фокус, приоритеты текущей проектной деятельности. На каждом последующем этапе «глубина» деятельности разработчиков увеличивается, охватывая большее количество функциональных характеристик и свойств системы. Например, на ракурсе развертывания проектные действия выполняются по всем стратам: от целевой страты до страты аппаратных средств (см. рис. 6).

Однако, не смотря на преимущества рассматриваемых подходов, в рамках которых предлагаются практические подходы к декомпозиции IT-систем отдельной предметной области и высокоуровневое представление жизненного цикла IT-проектов, необходимо отметить, что в данных концепциях отсутствуют подробные рекомендации по организации соответствующих им процессов разработки и внедрения.

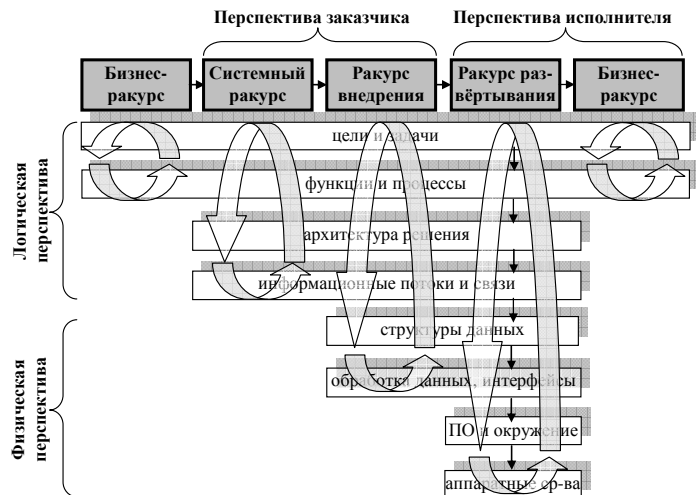


Рис. 6. Изменение фокуса процессов создания IT-систем в течение жизненного цикла разработки

Кроме того, концепция жизненного цикла NGOSS, к сожалению, рассматривает процесс развития системы без явно выделенного периода эксплуатации, подразумевая, что этот этап соответствует бизнес ракурсу. В то же время, развитие информационной системы продолжается и в процессе эксплуатации. Для телекоммуникационной отрасли, как уже отмечалось выше, этот фактор является очень существенным.

В ряде рекомендаций по управлению эксплуатацией корпоративных IT-систем и сервисов, например CobIT [10] выделяется отдельный этап

сопровождения. С точки зрения разработки ИТ-систем, эксплуатируемых в телекоммуникационной отрасли, данный этап обычно является продолжением развития в виде множества проектных активностей с меньшим масштабом, глубиной и длительностью, направленных на улучшение характеристик и функций системы, адаптацию и приспособление к текущим требованиям. Как правило, заказчиком формируется множество запросов на изменения (Change Request), которые оформляются исполнителем как мини проекты. В соответствии с декомпозицией ИТ-системы телекоммуникационного предприятия, показанной на Рис. 3, на этапе эксплуатации подобные проектные действия сфокусированы на уровнях адаптации и специализированных процессов. Действия, связанные с изменениями на уровнях общих сервисов, программно-аппаратных комплексов, относятся к фундаментальным проектам и программам модернизации или начальной разработки информационной системы.

Таким образом, процессы разработки, внедрения и эксплуатации ИТ-систем в соответствии с положениями NGOSS можно представить как последовательность действий в соответствии с ракурсами бизнеса, системы, внедрения и развертывания, но с определенным для каждого ракурса составом рассматриваемых уровней функциональной архитектуры. На рис. 7 показана подобная последовательность проектных действий для разработки внедрения ИТ-системы и последующей поддержки эксплуатации.

В соответствии со сделанным предположением о различных требованиях, предъявляемых к развитию ИТ-систем на различных стадиях жизненного цикла, рассмотрим особенности соответствующих ИТ-проектов, выполняемых на этих этапах. В табл. 2 показано сопоставление характеристик ИТ-проектов, выполняемых на двух основных стадиях жизненного цикла информационной системы телекоммуникационного предприятия.

Несмотря на явные различия характеристик проектов на стадиях разработки и эксплуатации системы, необходимо отметить, что спектр затрагиваемых свойств системы в каждом из подобных проектов остается широким и зависит от масштаба и глубины выполняемых действий на стадиях разработки или изменений на стадии эксплуатации ИТ-системы соответственно. Например, небольшие изменения на функциональном уровне адаптации системы могут повлечь изменения свойств и характеристик этого уровня, соответствующих не только стратам целей, функций, но и стратам интерфейсов и алгоритмов, программных средств, аппаратных ресурсов. Эта особенность проиллюстрирована на рис. 7, где показана «глубина» проектных действий относительно страт представления ИТ-системы.

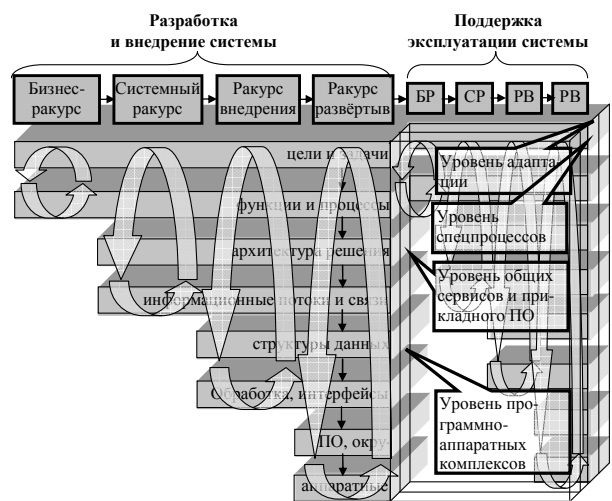


Рис. 7. Действия по развитию ИТ-системы на этапах разработки, внедрения и поддержки эксплуатации

Таблица 2- Характеристики проектов, выполняемых на основных стадиях жизненного цикла информационной системы

Характеристики проектов	Стадия разработки и внедрения	Стадия поддержкиэксплуатации
Масштаб	От проектов среднего до очень большого масштаба. Затрагиваются все функциональные уровни системы, часть компонентов проектируется и разрабатывается заново.	От проектов очень небольшого до среднего масштаба. Как правило, затрагиваются функциональные уровни, отвечающие за взаимодействие с пользователями, внешними подсистемами, иногда – за специализированную обработку данных
Длительность	От нескольких месяцев до нескольких лет	От нескольких дней до нескольких недель.
Требования к качеству	Высокие требования к качеству, обусловленные большим масштабом изменений и уровнем влияния на существующее окружение	Средние и невысокие требования в виду небольшого масштаба изменений и ограниченного влияния на окружение и на выполняемые функции в результате каждого отдельного изменения
Требования к преемственности	Необходимо обеспечить устойчивое и стабильное развитие на основе использования открытых расширяемых решений, создания артефактов, подробно описывающих разработанные компоненты. Результаты данной стадии являются основой для последующего развития	Как правило, вносимые изменения являются локальными, дальнейшее развитие может предполагать полную замену внесенных изменений
Периодичность	Раз в несколько лет	До нескольких мини-проектов в месяц

Сформулированные особенности проектов разработки на разных стадиях жизненного цикла информационной системы позволяют определить требования к методике управления соответствующими проектами.

### **Возможности использования методологий RUP и Agile на различных этапах разработки IT-систем**

С учетом особенностей методологий управления разработкой программной продукции, перечисленных в табл. 1, определим преимущества и недостатки RUP и Agile-подобных подходов применительно к проектам различного масштаба и длительности (см. табл. 3).

На основе исследования преимуществ и недостатков методологий управления разработкой программной продукции, требований и ограничений проектов на различных стадиях жизненного цикла информационных систем можно сформулировать следующие рекомендации по применению:

- подходы RUP имеет смысл применять для управления крупными, долгосрочными проектами уровня предприятия со средней и высокой сложностью;

- методология Agile удобна для быстрой разработки небольших задач и проектов с нечетко определенными объемами и сроками.



Таблица 3.Преимущества и недостатки RUP и Agile-подобных методологий при выполнении проектов различного масштаба и длительности.

		RUP	SCRUM
Длительные масштабные проекты	Преимущества	Детальное планирование проекта. Создание артефактов (документации), описывающих цели и результаты проекта. Всеобъемлющие процедуры тестирования, сдачи и приемки решений на этапе внедрения.	
	Недостатки	Необходимость привлечения выделенных высококвалифицированных специалистов различной специализации (аналитики, тестировщики, интеграторы), которые выполняют задачи не на постоянной основе, а в соответствии с планом проекта.	Короткий этап планирования проекта без детальной проработки решения. Планирование основывается на высокоуровневой декомпозиции на отдельные блоки и не предполагает предварительного анализа требований, что повышает риски невыполнения проекта
Краткосрочные, небольшие проекты	Преимущества		Высокая эффективность использования ресурсов, выполнение разработки от начала до конца одними исполнителями. Отсутствие длительных подготовительных процедур. Разбиение проекта на несколько этапов минимизирует риски несоответствия по качеству.
	Недостатки	Высокий уровень затрат на вспомогательные операции, не оправданный для небольших проектов. Артефакты, сопровождающие создание программной продукции являются излишними.	Высокие требования к квалификации исполнителей, необходимость владения несколькими специализациями. Разбиение проекта на этапы без детального планирования увеличивает риски несоответствия по времени и стоимости.

На основании сопоставления данных выводов и данных таблицы 1 (например, с использованием принципов SWOT), сформулируем выводы о предпочтительном применении методологий управления разработкой программных продуктов на различных этапах жизненного цикла IT-проекта (см. табл. 4). Необходимо отметить, что принятые в методологии RUP этапы разработки Inception, Elaboration, Construction и Transition во многом согласовываются с ракурсами жизненного цикла разработки внедрения IT-систем NGOSS: бизнес, система, внедрение и развертывание, соответственно. Особенностью является параллельное использование двух методологий на этапе внедрения. На этом этапе жизненного цикла информационной системы определяются объемы необходимой поддержки, желаемой модернизации, планируются итерации по развитию на конкретный период времени (например,

календарный год).

Таблица 4. Методологии управления созданием программных продуктов и жизненный цикл разработки и развития ИТ-систем

Жизненный цикл ИТ системы	Применяемые методологии	
	RUP	Agile-подобные
Планирование и анализ требований (бизнес-ракурс)	Inception	
Проектирование (системный ракурс)	Elaboration	
Реализация (ракурс имплементации)	Construction	
Внедрение (ракурс развертывания)	Transition	Предварительное проектирование
Поддержка эксплуатация (мн-во небольших изменений)		Последовательность итераций

Предложенная в табл. 4 схема является одним из возможных подходов к управлению развитием информационных систем на основе нескольких методологий, которые на первый взгляд не совсем совместимы, но в то же время, являются наиболее эффективными на различных этапах жизненного цикла.

Таким образом, предложен общий подход к формированию комплекса методологий управления ИТ-проектами в телекоммуникационной отрасли, для которой характерно наличие нескольких противоречивых фундаментальных требований к развитию OSS/BSS:

1) Определение основных противоречий в процессах развития ИТ-проектов.

2) Декомпозиция процессов развития ИТ-проектов в соответствии с жизненным циклом NGOSS на компоненты (этапы), в которых данные противоречия локализованы и не являются существенными.

3) Определение возможностей и ограничений, предъявляемых на каждом из этих этапов ИТ-проекту.

4) Сопоставление преимуществ и недостатков известных методологий с полученными возможностями и ограничениями процессов на каждом их этапов ИТ-проекта. На этом шаге возможно применение различных подходов к анализу преимуществ и недостатков, возможностей и ограничений, например SWOT.

5) Формирование, на основании выполненного анализа, комплекса методик, последовательно применяемых для управления ИТ-проектами, на каждом этапе жизненного цикла системы.

### **Выводы**

В работе сформулирован общий подход к применению нескольких методологий управления разработкой ПО в рамках проектов развития ИТ-систем. В качестве базовой концепции жизненного цикла информационной систем выбрана рекомендация NGOSS, дополненная промежуточным этапом сопровождения между циклами развития.

Показана возможность удовлетворения изначально противоречивых требований, предъявляемых к ИТ-проектам телекоммуникационной предприятия,

на основе декомпозиции функциональной архитектуры и перенесения основной нагрузки по выполнению небольших задач по адаптации и изменений интерфейсов и бизнес-процессов на отдельные функциональные уровни. Это, в свою очередь, позволяет использовать методологию RUP для IT-проекта разработки и внедрения, и методологии семейства Agile для управления мини-проектами изменений на этапе сопровождения эксплуатации. Также показано, что независимо от рассматриваемого функционального уровня на каждом из этапов в IT-проектах могут затрагиваться все группы свойств и характеристик: от целей системы до используемых программных и аппаратных средств.

К преимуществам такого подхода относится использование известных методологий управления разработкой, которые уже сопровождаются необходимыми инструментальными средствами, навыками и знаниями исполнителей. К недостаткам следует отнести сложности при переходе от одной методике к другой в последовательности IT-проектов, относящихся к разработке, внедрению и поддержке эксплуатации одной и той же информационной системы.

В рамках предложенного подхода в дальнейшем целесообразно разработать классификацию известных методологий управления разработкой программных продуктов проектами с целью формирования универсальных рекомендаций по формированию комплекса наиболее оптимальных методов управления отдельно взятыми IT-проектами.

**Список литературы:** 1. CobiT Mapping: Mapping of PMBOK with CobiT 4.0 / – USA: IT Governance Institute, 2006. – 44 p. 2. A guide to the project management body of knowledge. (PMBOK Guide) [текст] : Fourth Edition. – USA : PMI, Inc., 2008. – 467 p. 3. Pristley, M. A Unified Process for Software and Documentation Development / Michael Pristey, Mary Hunter Utt – 0-7803-6431-7/00/\$10© 2000 IEEE, 2000. – pp. 221-238. 4. Cobb, C.G. Making Sense of Agile Project Management / Charles G. Cobb – NJ: John Wiley & Sons Inc., 2011. – 245 p. 5. Udo, N. Will Agile Development Change the Way We Manage Software Projects? Agile form PMBOK® Guide Perspective [электронный ресурс] / Natalie Udo, Sonja Koppensteiner – ProjectwaybLLC, 2002 – Режим доступа : \WWW/ URL: <http://citeseerx.ist.psu.edu/viewdoc/download?0.1.1.125.3.pdf&ei=QK4nT-2dE8vktQbTw8G4AQ&usg=AFQjCNESlOqJVku-aypO2AUtmQ7JkjmD5g>. 6. What is the difference between RUP and SCRUM methodologies? [электронный ресурс] / Chiron Professional Journal – Режим доступа : \WWW/ URL: <http://www.chiron-solutions.com/chiron-professional-journal/2010/12/20/what-is-the-difference-between-rup-and-scrum-methodologies>. 7. Mohagheghi, P. The Impact of Software Reuse and Incremental Development on the Quality of Large Systems: Doctoral Thesis / Parastoo Mohagheghi – Trondhei : Norwegian University of Science and Technology, 2004. – 272 p. 8. *Илюшко, В. М.* Модели и методы информационной технологии проектирования метасистем [рук.] : дис. д-ра техн. наук: 05.13.06 / *В. М. Илюшко*. – Харьков : ХАИ, 1998. – 451 с. 9. New Generation Operational Support Systems (“NGOSS”). Architecture Overview. Public Version 1.500 [Электронный ресурс] / The TeleManagement Forum. – 2003. – Режим доступа : \WWW/ URL : <http://www.tmforum.org/sdata/documents/TMFC763%20GB920v1.5.pdf> – 10.11.2011 г. – Загл. с экрана. 10. CobiT 4.1: Framework, Control Objectives, Management Guidelines, Maturity Models / – USA: IT Governance Institute, 2007. – 196 p.

*Поступила в редколлегия 26.06.2012*