

Розглядаються питання наступності розвитку ІТ-систем. На основі декомпозиції на логічні і фізичні аспекти сформульовано принципи сталого і планомірного розвитку ІТ-систем

Ключові слова: наступність розвитку, ІТ-система, ризики, страта проектування

Рассматриваются вопросы преемственности развития ИТ-систем. На основе декомпозиции на логические и физические аспекты сформулированы принципы устойчивого и планомерного развития ИТ-систем

Ключевые слова: преемственность развития, ИТ-система, риски, страта проектирования

The questions of violation of continuity in development of it-systems are considered. The principles of sustainable and balanced development of the IT-systems are formulated based on the decomposition on the logical and physical aspects

Keywords: continuity of development, IT-system, risks, design stratum

МИНИМИЗАЦИЯ РИСКОВ НАРУШЕНИЯ ПРЕЕМСТВЕННОСТИ В РАЗВИТИИ ИТ-СИСТЕМ

А. В. Калмыков

Кандидат технических наук, докторант
Кафедра производства радиоэлектронных систем
летательных аппаратов
Национальный аэрокосмический университет им.
Н.Е. Жуковского «ХАИ»
ул. Чкалова, 17, г. Харьков, Украина, 61070
Контактный тел.: 067-570-05-36
E-mail: avk2007@list.ru

Введение

Жизненный цикл современных информационных систем предполагает длительные сроки эксплуатации, в течение которых возможна миграция на новые версии решения, расширение, замена существующей функциональности. В случае радикальных изменений или полной замены решения возможны проблемы с обеспечением стабильности используемых процедур и процессов, совместимости с другими компонентами программного и аппаратного окружения. При изменении какого-либо внутреннего компонента ИТ-системы возникает комплекс вопросов, связанных с разработкой, внедрением, развертыванием. Разработка и внедрение новых решений, миграция на связанные с этим измененные технологические и производственные процессы, новые организационные и административные политики, структуры и форматы используемых данных является очень затратной и болезненной процедурой [1].

Поэтому решение задачи обеспечения планомерного и устойчивого развития сложных систем, в том числе информационных, является актуальным и востребованным.

Постановка задачи

Эффективным подходом, который широко применяется в сложных технических решениях для их эволюционного развития, считается повторное использование успешных технических решений, процедур и артефактов. Для ИТ-систем преемственность выражается в следовании устоявшимся процессам, процедурам, интерфейсам и протоколам внешнего информационного обмена. В контексте длительных многоэтапных программ развития систем под пре-

емственностью понимается связь между процессами и объектами в развитии систем, при котором сохраняются наиболее удачные и эффективные элементы, подходы, процедуры и процессы.

Однако, полное следование консервативным принципам в развитии систем противоречит необходимости внедрения новой функциональности, обеспечения соответствия фактическим изменениям внешнего окружения [2].

Следовательно, в жизненном цикле ИТ-систем существует противоречие между необходимостью развития функциональности и обеспечением стабильности существующих процедур и компонентов. В связи с этим возникает потребность в определении принципов организации процессов проектирования, разработки, внедрения ИТ-систем, позволяющих разрешить это противоречие. Как одно из направлений исследований этой проблематики в рамках данной статьи рассматриваются вопросы влияния различных рисков на стабильность процессов и артефактов разрабатываемых ИТ-систем.

Известные подходы к разработке и внедрению ИТ-систем

Подходом, учитывающим особенности жизненного цикла ИТ-систем, считается методология TMF NGOSS [3, 4], применяемая для управления развитием производственных структур телекоммуникационной отрасли. При этом развитие ИТ-инфраструктуры рассматривается с учетом позиций внутренних и внешних сторон, особенностей реализации решений на каждом этапе жизненного цикла. На рис. 1 показано схематическое изображение ракурсов жизненного цикла системы согласно методологии NGOSS. Под ракурсом жизненного цикла в данном случае

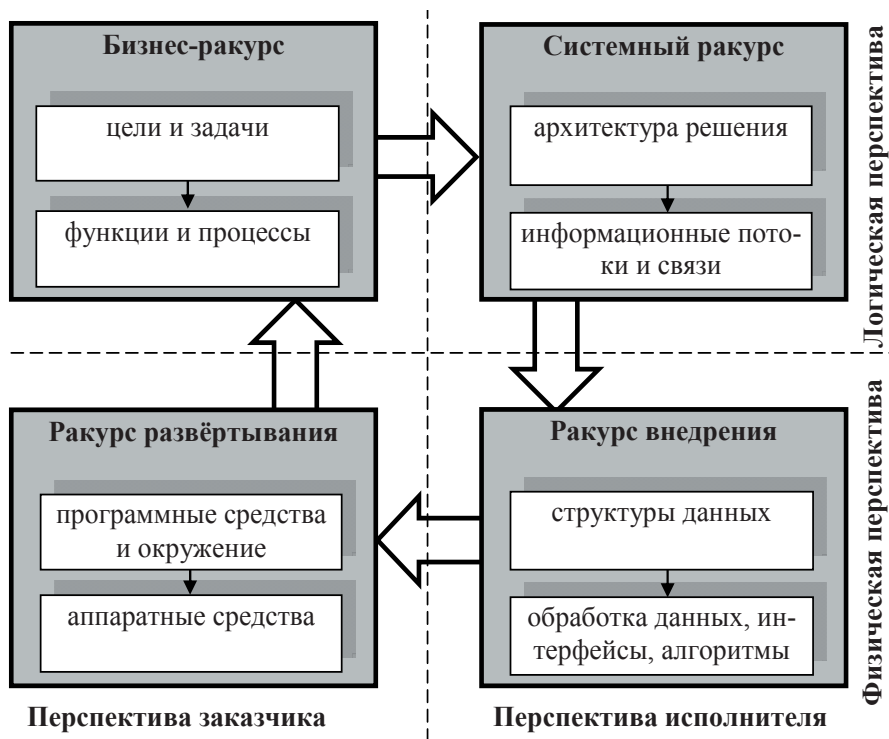


Рис. 1. Жизненный цикл IT-систем в соответствии с TMF NGOSS

понимаются не столько конкретные действия по анализу, разработке, внедрению, развертыванию решения, сколько фокус деятельности исполнителя и заказчика, их приоритеты и взгляды на выполнение соответствующих задач [5].

Основываясь на этом, возможно выделить технологически независимые аспекты (логическая перспектива) IT-систем и технологически зависимые (физическая перспектива) компоненты. Очевидно, что изменения в технологически зависимых компонентах является наиболее затратными по отношению непосредственно к IT-системе, в то время

как изменения в логических аспектах являются наиболее болезненным по отношению к внешнему окружению системы в случае, если они выполняются не в соответствии с его требованиями. Следовательно, одним из возможных условий обеспечения эффективного развития может являться концентрация потенциальных изменений в аспектах, относящихся к логической перспективе жизненного цикла системы.

Вопросы проектирования и разработки сложных систем на основе различных подходов к декомпозиции их свойств и функций исследованы в работах многих авторов [6-9]. При этом предлагается декомпозиции процесса разработки и разрабатываемой системы на страты, соответствующие различным уровням свойств и характеристик информационной системы. В работах [6, 7] предложено вы-

полнять разработку сложных информационно-компьютерных систем последовательно по таким стратам: целевая, функциональная, информационная, структурная, данных, алгоритмическая, ПО, КТС.

На рис. 2 показаны позиции страт IT-системы относительно ракурсов жизненного цикла NGOSS. Логическое проектирование выполняется по стратам верхних уровней, и его результаты учитываются для нижестоящих страт IT-системы, на каждой последующем страте разработки происходит уточнение и детализация разрабатываемой системы, принимаются

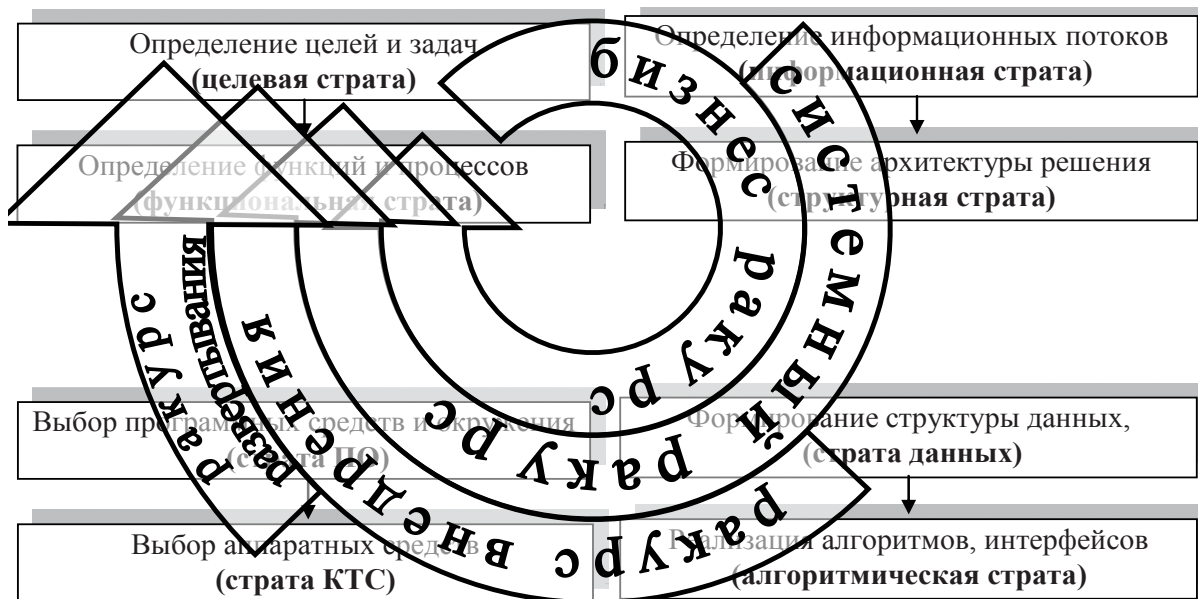


Рис. 2. Стратификация IT-систем и ракурсы жизненного цикла

во внимание факторы, соответствующие актуальным для данного этапа ракурсам жизненного цикла.

На основе циклического, повторяющегося характера выполняемых действий, как показано на рис. 2, и с учетом особенностей отдельных страт проектирования, становится возможным описать принципы и специфику развития ИТ-систем для задач обеспечения преемственности.

Принципы преемственности развития ИТ-систем

Запишем несоответствие между наборами требуемых характеристик $R_{stra}^{i,ref}$ и начальным состоянием системы $R_{stra}^{0,ref}$ для страты $stra$ для i -го цикла развития системы как:

$$\Delta R_{stra}^i = \overline{R_{stra}^{i,ref} \cap R_{stra}^{0,ref}} = (R_{stra}^{i,ref} \setminus R_{stra}^{0,ref}) \cup (R_{stra}^{0,ref} \setminus R_{stra}^{i,ref}), \quad (1)$$

где $R_{stra}^{i,ref} \setminus R_{stra}^{0,ref}$ – новый набор требований, которым система не удовлетворяет;

$R_{stra}^{0,ref} \setminus R_{stra}^{i,ref}$ – существующий набор характеристик, которые не соответствуют $R_{stra}^{i,ref}$, из дальнейшего рассмотрения исключается.

Из рис. 3 следует, что с учетом цикличности развития системы набор желаемых характеристик на страте $stra$ на $i+1$ -м цикле развития системы запишется как:

$$R_{stra}^{i+1,ref} = RC_{stra}^{i+1} \cup (R_{stra}^{i,ref} \cap R_{stra}^{i+1,ref}), \quad (2)$$

где RC_{stra}^{i+1} – совокупность новых требований и ограничений, предъявляемых к системе в $i+1$ -м цикле развития.

Различие между характеристиками идеального состояния системы на $i+1$ -м цикле развития системы и начального состояния $R_{stra}^{0,ref}$ будет определено как:

$$\begin{aligned} \Delta R_{stra}^{i+1} &= R_{stra}^{i+1,ref} \setminus R_{stra}^{0,ref} = \\ &= (RC_{stra}^{i+1} \cup (R_{stra}^{i,ref} \cap R_{stra}^{i+1,ref})) \setminus R_{stra}^{0,ref} = \\ &= RC_{stra}^{i+1} \setminus R_{stra}^{0,ref} \cup (R_{stra}^{i,ref} \cap R_{stra}^{i+1,ref}) \setminus R_{stra}^{0,ref} \end{aligned} \quad (3)$$

Очевидно, что обеспечение преемственности развития системы зависит от объема требований на изменение на каждом цикле развития, предъявляемых к системе. Чем меньший объем новых требований, тем проще обеспечить преемственность заложенных в систему решений и связанных с ними процедур обслуживания и сопровождения.

В зависимости от уровня преемственности набора предъявляемых к системе требований возможны следующие случаи:

- если обеспечивается полная преемственность с добавлением новых требований:

$$R_{stra}^{i,ref} \in R_{stra}^{i+1,ref} \Rightarrow R_{stra}^{i,ref} \cap R_{stra}^{i+1,ref} = R_{stra}^{i,ref}; \quad (4)$$

- если обеспечивается полная преемственность без добавления новых требований:

$$R_{stra}^{i,ref} \in R_{stra}^{i+1,ref} \Rightarrow R_{stra}^{i,ref} \cap R_{stra}^{i+1,ref} = R_{stra}^{i+1,ref}; \quad (5)$$

- если обеспечивается частичная преемственность:

$$\begin{aligned} R_{stra}^{i+1,ref} \cap R_{stra}^{i,ref} &\neq R_{stra}^{i,ref}, \quad R_{stra}^{i+1,ref} \cap R_{stra}^{i,ref} \neq R_{stra}^{i+1,ref}, \\ R_{stra}^{i+1,ref} \cap R_{stra}^{i,ref} &\neq \emptyset; \end{aligned} \quad (6)$$

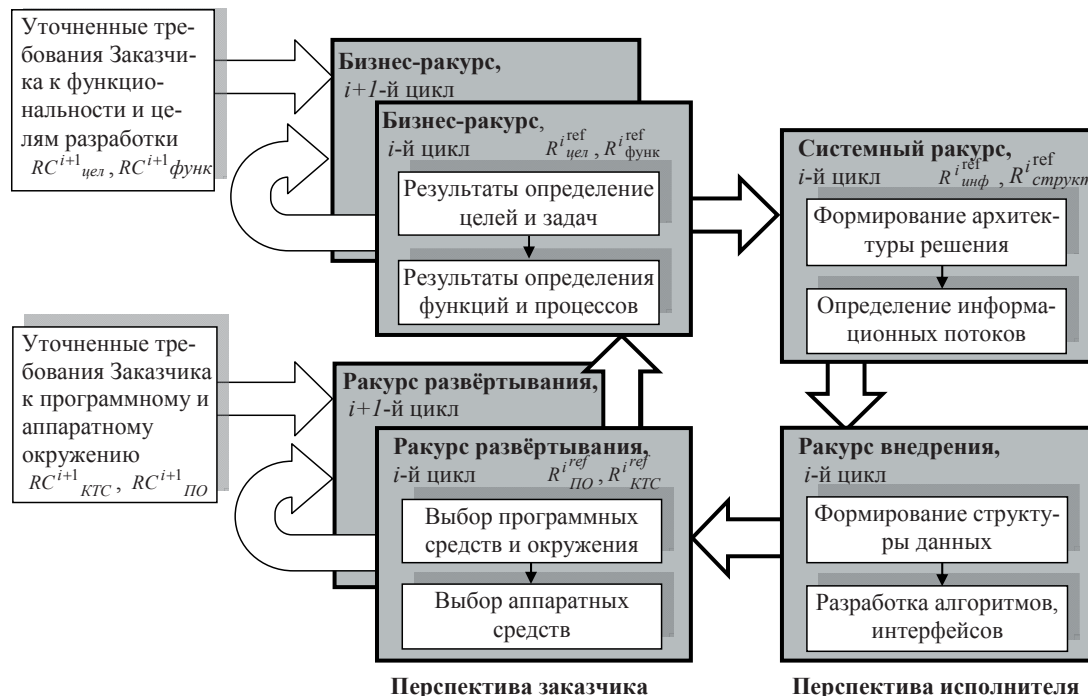


Рис. 3. Схема жизненного цикла ИТ-систем в многоэтапных проектах развития

- если преемственность не обеспечивается:

$$R^{i+1ref}_{stra} \cap R^{iref}_{stra} = \emptyset .$$

В качестве целевой функции для обеспечения преемственности системы выберем минимизацию сложности (стоимости) реализации требований на протяжении всей многоэтапной программы развития [10]. Стоимость реализации желаемых характеристик и требований к системе для $i+1$ -го цикла развития будет записана как:

$$C(\Delta R^{i+1ref}_{stra}) = C(RC^{i+1}_{stra} \setminus R^{0ref}_{stra}) + C((R^{iref}_{stra} \cap R^{i+1ref}_{stra}) \setminus R^{0ref}_{stra}) , \quad (7)$$

В то же время стоимость уже реализованных требований и характеристик на i -м цикле развития системы может быть записана как:

$$C(\Delta R^{iref}_{stra}) = C(((R^{iref}_{stra} \setminus R^{i+1ref}_{stra}) \cup (R^{iref}_{stra} \cap R^{i+1ref}_{stra})) \setminus R^{0ref}_{stra}) = C((R^{iref}_{stra} \setminus R^{i+1ref}_{stra}) \setminus R^{0ref}_{stra}) + C((R^{iref}_{stra} \cap R^{i+1ref}_{stra}) \setminus R^{0ref}_{stra}) , \quad (8)$$

где $[R^{iref}_{stra} \cap R^{i+1ref}_{stra}] \setminus R^{0ref}_{stra}$ – актуальные требования, реализованные на i -м цикле развития системы;
 $[R^{iref}_{stra} \setminus R^{i+1ref}_{stra}] \setminus R^{0ref}_{stra}$ – требования, реализованные на i -м цикле развития системы, но ставшие не актуальными на $i+1$ -м цикле;
 $RC^{i+1}_{stra} \setminus R^{0ref}_{stra}$ – требования, реализуемые на $i+1$ -м цикле развития системы.

Целевая функция (минимизация стоимости разработки и развития системы) запишется как:

$$\sum_{str} \left[\sum_i C(RC^i_{str} \setminus R^{0ref}_{str}) \right] \rightarrow \min \quad (9)$$

$$\begin{aligned} & \sum_{str} \left[\sum_i C(RC^i_{str} \setminus R^{0ref}_{str}) \right] = \\ & = \sum_{str} \left[\sum_i (C(\Delta R^{iref}_{str}) - C((R^{i-1ref}_{str} \cap R^{iref}_{str}) \setminus R^{0ref}_{str})) \right] = \\ & = \sum_{str} \left[\sum_i \left(C([R^{iref}_{str} \setminus R^{i+1ref}_{str}] \setminus R^{0ref}_{str}) + C([R^{iref}_{str} \cap R^{i+1ref}_{str}] \setminus R^{0ref}_{str}) - C((R^{i-1ref}_{str} \cap R^{iref}_{str}) \setminus R^{0ref}_{str}) \right) \right] \quad (10) \end{aligned}$$

Очевидно, что выполнения целей (10) необходимо выполнение условий:

$$\sum_{str} \sum_i (C([R^{iref}_{str} \setminus R^{i+1ref}_{str}] \setminus R^{0ref}_{str})) \rightarrow \min , \quad (11)$$

$$\sum_{str} \left[\sum_i (C([R^{iref}_{str} \cap R^{i+1ref}_{str}] \setminus R^{0ref}_{str}) - C((R^{i-1ref}_{str} \cap R^{iref}_{str}) \setminus R^{0ref}_{str})) \right] \rightarrow \min \quad (12)$$

Полагая, что $C(R)$ являются монотонной и возрастающей функцией, получаем из (11) и (12), соответственно:

$$\begin{aligned} & \bigcup_{str} \left([R^{iref}_{str} \setminus R^{i+1ref}_{str}] \setminus R^{0ref}_{str} \right) \rightarrow \min \Rightarrow \\ & \Rightarrow [R^{iref}_{str} \setminus R^{i+1ref}_{str}] \rightarrow \min \Rightarrow [R^{iref}_{str} \approx R^{i+1ref}_{str}] \quad (13) \end{aligned}$$

$$\begin{aligned} & \bigcup_{str} \left([R^{iref}_{str} \cap R^{i+1ref}_{str}] \setminus R^{0ref}_{str} \right) \setminus \left([R^{i-1ref}_{str} \cap R^{iref}_{str}] \setminus R^{0ref}_{str} \right) \rightarrow \min \Rightarrow \\ & \Rightarrow [R^{iref}_{str} \cap R^{i+1ref}_{str}] \setminus [R^{i-1ref}_{str} \cap R^{iref}_{str}] \rightarrow \min \quad (14) \end{aligned}$$

Однако, (13) и (14) фактически противоречат расширению требований, предъявляемых к системе, развитию ее функциональности. С другой стороны эти выражения можно трактовать как необходимость сохранения на текущих этапах развития максимального объема требований, предъявляемых к системе, начиная с первых итераций и на протяжении всего ее развития. В то же время, на начальное состояние системы R^{0ref}_{stra} не накладываются какие-либо ограничения.

Если рассматривать полный отдельный цикл разработки ИТ-системы, то в соответствии с процессами разработки сложных систем, обоснованными в работе [6], и показанной на рис. 3 схемой жизненного цикла, изменения, вносимые на страте $stra$, влияют на формирование набора требований на последующих стратегах и на соответствующие им процессы разработки:

$$RC^i_{stra} = R \left(\bigcup_{strx=F\bar{x}}^{stra-1} RC^i_{strx} \right) , \quad (15)$$

где

$$str \subset \left[\begin{array}{l} \text{целевая, функциональная, информационная,} \\ \text{структурная, данные, алгоритмическая, ПО, КТС} \end{array} \right]$$

Тогда стоимость разработки системы можно записать следующим образом:

$$\begin{aligned} & \sum_{str} \left[\sum_i C(RC^i_{str} \setminus R^{0ref}_{str}) \right] = \\ & = \sum_i \left[\sum_{str} C \left(R \left(\bigcup_{strx=F\bar{x}}^{str-1} RC^i_{strx} \right) \setminus R^{0ref}_{str} \right) \right] = \\ & = \sum_i \left[\sum_{str} C \left(R \left(\bigcup_{strx=F\bar{x}}^{str-1} [R^{iref}_{strx} \setminus (R^{i-1ref}_{strx} \cap R^{iref}_{strx})] \right) \setminus R^{0ref}_{str} \right) \right] \quad (16) \end{aligned}$$

Из (16) следует, что наибольшее влияние на разработку системы и, соответственно, на ее сложность и стоимость оказывают результаты проектных действий по стратегам верхних уровней.

Полагая, что функции $R(RC)$ и $C(R)$ являются монотонными и возрастающими, из (15) также можно сделать вывод о предпочтительности небольших, некардинальных изменений в требованиях и функциональности:

$$C(R) \rightarrow \min \Rightarrow R(RC) \rightarrow \min \Rightarrow$$

$$R \left(\bigcup_{strx=F\bar{x}}^{str-1} [R^{iref}_{strx} \setminus (R^{i-1ref}_{strx} \cap R^{iref}_{strx})] \right) \setminus R^{0ref}_{str} \rightarrow \min \Rightarrow$$

$$R_{strx}^{iref} \rightarrow \min \tag{17}$$

Условие (17) также противоречит необходимости развития и обновления системы.

Однако, имеется возможность рациональной организации процессов проектирования и развития, принимая во внимание следующие особенности процессов разработки информационных систем:

- наличие четко выраженной логической части процесса проектирования, которая не подразумевает физическую реализацию и относится, в-основном, к первым этапам;
- существенное влияние результатов именно первых этапов проектирования на стоимость всего процесса разработки, особенно на первых итерациях развития системы.

На основании данных особенностей сформулируем основные принципы минимизации рисков нарушения преемственности, устойчивости и планомерности развития ИТ-систем:

- проектирование системы с учетом максимального возможного объема перспективных требований и функциональности на стратах, соответствующих ракурсам логической перспективы, до начала физической реализации системы;
- проектирование системы по стратам проектирования, соответствующим только физической перспективе с учетом максимально возможного ис-

пользования существующих компонентов, реализованных на предыдущих циклах развития системы.

Риски нарушения преемственности и возможности их минимизации

В соответствии с полученными выводами задача максимального покрытия возможных требований при формировании начального состояния системы R_{stra}^{0ref} решается на стратах проектирования (целевой, функциональной, информационной, структурной), которые соответствуют логическим ракурсам жизненного цикла. При этом на практике, как правило, используются архитектурные решения, обеспечивающие максимальную независимость существующей функциональности от вновь реализуемой. На стратах, на которых проектные действия выполняются и оцениваются только с позиций физических ракурсов, преемственность в развитии обеспечивается применением стандартизированных решений и средств разработки. При этом решения, принимаемые на начальных итерациях, должны обеспечивать возможности реализации требуемой функциональности на последующих итерациях и учитывать потенциальные риски, которые могут нарушить ожидаемую преемственность и планируемые шаги по развитию информационной системы. В табл. 1 показаны возможные риски, способы их минимизации, стороны принимающие решения (СПР) относительно страт проектирования и ракурсов жизненного цикла ИТ-систем.

Таблица 1

Потенциальные риски развития информационных систем, возможности их минимизации и обеспечения преемственности разрабатываемых решений

Страта	Ракурс	СПР	Возможные риски развития системы	Возможности минимизации рисков
Целевая	бизнес, системный, внедрение, развёртывание	Заказчик	Изменение целей и функциональности, которое может привести к несовместимости с существующими артефактами	Определение долгосрочных целей и детальная проработка функциональности с определением очередности их реализации
Функциональная				
Информационная	системный, внедрение (реализация), развёртывание	Исполнитель	Взрывной рост объемов, усложнение связей информационного обмена	Использование архитектуры, позволяющей унифицировать и масштабировать обмен информацией
Структурная			Необходимость внедрения новой функциональности, не совместимой с существующими решениями, увеличение объемов обработки данных приводит к возникновению «узких мест»	Использование архитектуры, допускающей масштабирование производительности и наращивание функциональности с сохранением необходимой существующей
Данные	внедрение (реализация), развёртывание	Исполнитель	Радикальное изменение форматов данных, условий их хранения и обеспечения доступа	Применение структур данных, допускающих расширение (в некоторых случаях заведомо избыточные решения), инструментов, позволяющих выполнять миграцию на новые структуры.
Алгоритмическая			Кардинальное изменение логики обработки данных	Использование case- инструментов разработки и сопровождения, алгоритмов и процедур обработки данных, подходов, ориентированных на повторное использование компонентов (MDA, SOA), разделение бизнес-логики и функционального уровня обработки данных
ПО	развёртывание	Заказчик	Переход на другое базовое или системное программное обеспечение	Использование кросс-платформенного прикладного программного обеспечения
КТС			Переход на другое аппаратное обеспечение	Использование кросс-платформенного системного программного обеспечения

Данные рекомендации соответствуют полученным ранее выводам о принципах обеспечения преемственности в развитии IT-систем.

Выводы

Таким образом, в данной статье сформулированы принципы обеспечения преемственности процессов и артефактов IT-систем в длительных многоэтапных программах развития. Показана целесообразность полноценных проектных действий на начальных стадиях и стратах проектирования, соответствующих логическим ракурсам, и перспективность итерацион-

ных, эволюционных изменений на этапах и стратах, соответствующих физической перспективе жизненного цикла.

На основе стратификации процессов разработке предложен подход к определению и декомпозиции потенциальных рисков нарушения преемственности в развитии информационных систем и рассмотрены возможности минимизации их влияния.

Дальнейшие перспективы развития исследований в рассматриваемой проблемной области состоят в формировании на основании полученных выводов свода практических рекомендаций и дополнений к используемым методикам управления проектами разработки, внедрения и развертывания IT-систем.

Литература

1. Mohagheghi, P. The Impact of Software Reuse and Incremental Development on the Quality of Large Systems: Doctoral Thesis / Parastoo Mohagheghi – Trondheim : Norwegian University of Science and Technology, 2004. – 272 p.
2. Bassam, A. Reusable Software Component Life Cycle / Anas Bassam AL-Badareen, Mohd Hasan Selamat, Marzanah A. Jabar, et.al – NAUN, International Journal of Computers, issue 2, vol.5, 2011. – pp.191-199.
3. GB 927. The NGOSS Lifecycle and methodology. Version 1.1 [Электронный ресурс] / The TeleManagement Forum. – 2004. – Режим доступа : \www/ URL: [http://www.itarchitects.ca/whitepaper/The%20NGOSS%20Lifecycle %20and%20 Methodology.pdf](http://www.itarchitects.ca/whitepaper/The%20NGOSS%20Lifecycle%20and%20Methodology.pdf) – 15.05.2010 г. – Загл. с экрана.
4. 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.
5. Райли, Дж. NGOSS: Построение эффективных систем поддержки поддержки и эксплуатации сетей оператора связи [текст] / Дж. Райли, М.Кринер. – пер. с англ. – М.: Альпина Бизнес Букс, 2007. – 192 с.
6. Илюшко, В. М. Системное моделирование в управлении проектами [текст] : монография / В. М. Илюшко, М. А. Латкин. – Харьков : НАУ «ХАИ», 2010. – 220 с.
7. Калмыков А.В., Особенности управления разработкой и внедрением информационных систем телекоммуникационной отрасли / А.В. Калмыков, А.А. Рева // Восточно-Европейский журнал передовых технологий. – 2011. – № 6/3(54). – с. 29-35.
8. Слюсаренко, М.Ю. Системологический подход к декомпозиции в объектно-ориентированном анализе и проектировании программного обеспечения [Электронный ресурс] / И.М. Слюсаренко, М.Ю Слюсаренко. – Режим доступа : \www/ URL : http://citforum.ru/programming/case/ood_systemology.
9. GB 921. Enhanced Telecom Operations MAP (eTOM). The Business Process Framework. TMF Approved Version 4.0 [Электронный ресурс] / The TeleManagement Forum. – 2004. – Режим доступа : \www/ URL: http://www.idef.ru/documents/tmfc2495_gb921_v4-0-1_040318.pdf – 15.05.2011 г. – Загл. с экрана.
10. Kontio, J. Defining Factors, Goals and Criteria for Reusable Components Evaluation /Jyrki Kontio, Gianluigi Caldiera, Victor R. Basili – Cascon Conference, Toronto Canada, November 14-16, 1996. – pp. 211-223.