

УДК 021.8 + 025.1
ББК 78.34

РИСКИ И ВЫБОР ОПТИМАЛЬНЫХ ПРОЕКТОВ: СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ

Пырлина И. В.¹

(ГУ Высшая школа экономики, Москва)

В статье предложен обзор известных моделей выбора проектов реализации информационных систем с сервис-ориентированной архитектурой (СОА), методов оценки рисков и многокритериального анализа информационных систем с различными типами архитектур, включая СОА. Статья также описывает предложенные методы анализа рисков от СОА и метод многокритериального анализа проектов информационных систем с СОА. Применение данных методов представлено с использованием прототипа инструмента, который состоит из двух подсистем – подсистема имитационного моделирования и подсистема многокритериального анализа.

Ключевые слова: сервис-ориентированная архитектура, операционные риски при сервис-ориентированном подходе, критерии эффективности архитектурного проекта, метод порогового агрегирования, ранжирование проектов.

1. Введение

В современных предприятиях сервис-ориентированная архитектура информационных систем (ИС) используется для создания более открытой и гибкой информационной среды предприятия, более сбалансированного ландшафта приложений

¹ Ирина Владимировна Пырлина, специалист, эксперт ИТ-архитектуры (irina.pyrlina@gmail.com).

без дублирования функциональности. Под функциональностью в данном случае понимается совокупность решаемых задач в рамках информационной системы, предоставленных пользователю в виде вызываемых функций. Она позволяет адаптировать инновации, среди которых мобильные технологии и «облачные» системы. Проблема выбора проекта внедрения информационной системы с сервис-ориентированной архитектурой становится более актуальной с появлением различных вариантов ее реализации от поставщиков решений.

Под проектом [6] понимается ограниченное во времени целенаправленное изменение отдельного приложения или информационной системы, а также реализация новой информационной системы с сервис-ориентированной архитектурой с определенными целями, достижение которых означает завершение проекта, а также с установленными требованиями к результату, срокам, рискам, бюджету и ресурсам. Фокусируясь больше на технической стороне реализации COA, предприятия пренебрегают другими, не менее важными аспектами этой концепции. Однако успешность проекта реализации COA зависит от учета всех аспектов готовности организационной архитектуры, специалистов и механизмов управления. Поэтому выбор проекта реализации ИС с сервис-ориентированной архитектурой на предприятии должен учитывать эти немаловажные аспекты.

В данной работе предлагается метод многокритериального анализа информационных систем с сервис-ориентированной архитектурой, который должен использоваться любым предприятием, принимающим решения по корпоративной архитектуре и планирующим портфель ИТ-проектов. Несмотря на то, что мы находимся в начале периода повсеместного использования новой архитектуры, если верить Gartner [29], в ландшафте большинства компаний все еще есть разнообразие систем и архитектур. Причинами отказа от перехода на архитектуру COA, как показано в [29], являются в первую очередь недостаток экспертизы COA, отсутствие или непонимание выгод для бизнеса, стоимость и недостаток необходимых для внедрения ресурсов, рискованность перехода

на относительно новую концепцию, сложность архитектуры, и отсутствие необходимости. Поэтому важным вопросом сегодня является разработка методики выбора проектов реализации информационных систем с сервис-ориентированной архитектурой, также как и обоснование эффективности и рисков решения. Рассмотрим предлагаемые подходы оценки рисков и выбора оптимального проекта.

2. Выбор оптимального проекта информационной системы с сервис-ориентированной архитектурой

Прежде чем перейти к определению методов выбора проектов, введем несколько ключевых понятий. Под информационной системой [7] полагаются «аппаратно-программные системы, которые поддерживают приложения с интенсивной обработкой данных». При этом приложением информационной системы (или прикладной программой) в данном случае считается «надстройка над информационной системой, обеспечивающая решение некоторого комплекса задач».

Информационная система также может являться компонентой (или подсистемой) более сложной информационной системы [7]. Например, информационная система развития и управления персоналом и информационная система бухгалтерского учета являются компонентами информационной системы предприятия.

Информационные системы используют несколько ресурсов [7], к ним относятся средства вычислительной техники, системное и прикладное программное обеспечение, информационные, лингвистические и человеческие ресурсы.

Прикладные ресурсы (или прикладное программное обеспечение) – прикладные программы, которые имеют возможность публикации/поддержки технических сервисов; к их числу относятся не только программное обеспечение (ПО), но и системы аппаратного обеспечения. Прикладное ПО [7] используется для решения конкретного класса задач и обычно в

качестве таких средств используются коммерческие программные продукты: СУБД общего назначения, Web-серверы, системы управления документами, информационные системы, имеющие узкое назначение (используемые для поддержки конкретных процессов предприятия).

Под системой управления базой данной (СУБД) понимается база данных со специализированным программным обеспечением, которое позволяет структурированно хранить и обрабатывать данные [6].

Подсистема доступа – различные технологии пользовательского интерфейса для работы с системой («тонкий» клиент на базе Web-технологий и «толстый» клиент в виде прикладных Windows-приложений). Пользовательские ресурсы [7] в интернете – это, например, страницы Web-сайтов, разные доступные пользователям Web-документы, страницы HTML и другие форматы интерфейса.

Инфраструктурное программное обеспечение (или системное) включает [7] различные операционные системы, системные оболочки, служебные программы для поддержки серверного оборудования и работы администратора, сетевое программное обеспечение и телекоммуникационное оборудование.

Кроме того, для успешного оперирования информационной системы необходимы и технические ресурсы, к которым относятся сервера, помещение, электричество и другие.

2.1. МЕТОД КЛАССИФИКАЦИИ И ОЦЕНКИ РИСКОВ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ

Операционные риски¹ в рамках информационных систем с сервис-ориентированной архитектурой тесно связаны с ошибками работы ресурсов, потребляемых этими приложениями, в частности с ресурсами бизнес-сервисов.

¹ *Общее определение термина «операционный риск» приводится в [13, стр.478].*

Оценка рисков ИС с сервис-ориентированной архитектурой производится на базе статистики ошибок, собранной по анализируемым проектам. Качество информационной системы напрямую выражается через количество ошибок в различных компонентах системы, соответственно и риски ИС с сервис-ориентированной архитектурой представляются в виде рисков возникновения ошибок компонентов информационной системы. Используем следующую классификацию рисков ИС с сервис-ориентированной архитектурой:

1. Риски ввода-вывода результатов.
2. Функциональные риски.
3. Риски связующего программного обеспечения (ПО).
4. Риски обработки данных.
5. Системные риски.

При классификации ошибок мы будем также указывать приоритет ошибки (1 – очень высокий, 2 – высокий, 3 – средний, 4 – низкий) и сложность устранения ошибки или длительность рассмотрения, так как то, как быстро была решена проблема, характеризует сложность проблемы. Длительность рассчитывается от даты подачи заявления до даты, когда проблема считается решенной, т.е. выставлен статус «completed». Обоснование такой классификации рисков приводится в [13].

Предлагается использовать риски как отдельную группу критериев, влияющих на выбор проекта реализации ИС с сервис-ориентированной архитектурой. Для проверки предложенной классификации было проанализировано 4435 ошибок по семи проектам за период с 2006-2010 гг. Все ошибки разделены на четыре приоритета по степени убытка и важности их устранения («очень высокий» – «высокий» – «средний» – «низкий»). Собранная статистика удовлетворяла нескольким типам функции распределения в зависимости от типа ошибок. В случае типов ошибок ввода-вывода результатов, ошибок данных и ошибок связующего звена четко определяется экспоненциальное распределение, что видно и по графикам функций вероятностей (рис 1). В то время как системные

ошибки имеют распределение Пуассона. А функциональные ошибки имеют нормальное распределение.

Под приоритетом понимается серьезность наносимого убытка работе информационной системы и предприятию в целом при возникновении ошибки. Предлагается различать 4 типа приоритетов:

1. Очень высокий – ошибки, которые имеют или оказывают критическое влияние на продуктивную эксплуатацию системы, на процессы и операции предприятия, а также могут привести к краху или полной остановке работы системы.

2. Высокий приоритет имеют ошибки, которые оказывают существенное влияние на процессы и операции предприятия, а также ведут к заметному снижению производительности системы.

3. Средний приоритет имеют ошибки, которые оказывают влияние на работу системы, и соответственно могут повлиять на процессы и операции, однако не являются критическими.

4. Низкий приоритет получают ошибки, прозрачные для пользователя. Например, ошибки пользовательского интерфейса не нарушают возможность использования системы, однако делают использование некомфортным и требуют устранения.

В рамках данного анализа применение метода порогового агрегирования для системы с большим числом приоритетов (типов ошибок по рискам) нецелесообразно. А преимущества максиминного метода (или метода Гермейера) по сравнению с методом линейной свертки приведены в [9]. Поэтому для ранжирования данных по рискам была применена максиминная свертка (1). Определим $N = \{n_1, n_2, \dots, n_k\}$ – множество сообщений об ошибках, тогда агрегирование рисков происходит по следующему правилу:

Определим матрицу S^+ такую что,

$$(1) \quad \forall x \in T, \forall y \in P, S^+ = \{n(x, y)\},$$

где T – множество типов рисков; P – множество приоритетов; R – пространство рангов; $x_i, i \in [1, |T|]$, – количество ошибок по типам рисков.

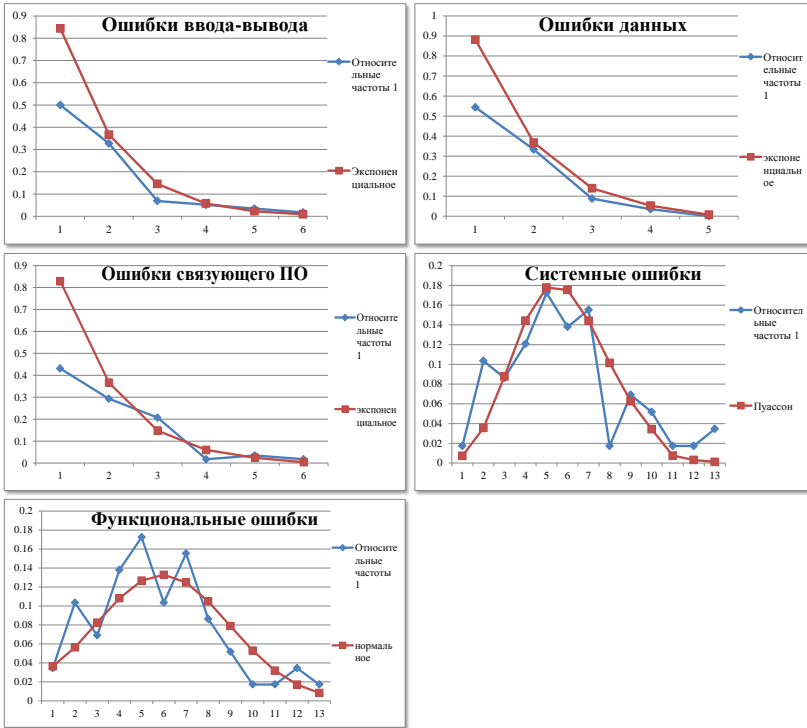


Рис. 1. Плотности вероятностей возникновения по типу рисков

В матрице S^+ на пересечении строки x и столбца y стоит число $n(x, y)$, которое равно количеству ошибок типа x по приоритету y . Выберем минимальное значение в каждой строке матрицы S^+ . А затем выберем наибольшее значение из выбранных. Результирующее «максиминное» значение (т.е. группа рисков) получает максимальный ранг. Далее повторим процедуру для оставшихся групп рисков. Таким образом, правило выглядит так: x признается лучшим и выбирается тогда и только тогда, когда

$$a(x) = \min_{b \in P} \{n(x, b)\},$$

где $a(x)$ – оценка, соответствующая риску x , которая позволяет определить упорядоченное отображение из множества типов рисков во множество рангов

$$(2) \quad T \rightarrow R: r[a(x)] \in \{\overline{1,4}\},$$

$$\text{такое, что } r[a(x)] = \begin{cases} 1, \max_{x \in T} \{a(x)\}; \\ 2, \max_{x \in T \setminus \{x_1\}} \{a(x)\}; \\ \dots \end{cases}$$

где r из R – ранг соответствующий некоторому типу рисков x .

Таким образом, в результате применения максиминного метода получим вектор значений с рангами. В случае максиминного критерия получается гарантированная нижняя оценка для всех $n(x, y)$ [9].

Таким образом, в данном случае $r(x_i)$ представляет собой вектор, значения которого приобретают ранг 1, 2, 3 или 4 в зависимости от того, какое из значений матрицы S^+ становится минимальным по приоритетам ошибок и одновременно максимальным по типам ошибок.

Переход от числовых оценок к рангам необходим для приведения результатов анализа рисков к виду, позволяющему сопоставление их с общими требованиями многокритериального анализа вариантов проектов по методу порогового агрегирования.

На основе метода перехода от числовых оценок к рангам самый высокий ранг получают функциональные риски для обоих проектов, описанный в [12]. При ранжировании были выбраны самые высокие из минимальных оценок, что говорит о наибольшем количестве ошибок и соответственно свидетельствует о наибольшем риске в соответствующем типе рисков. В данном случае первый (самый высокий) ранг присваивается наибольшей оценке риска, так как чем больше риск, тем хуже предлагаемый проект. Поэтому, в отличие от ранжирования по всем остальным критериям, которые будут рассматриваться в дальнейшем, в случае с рисками используется противоположный метод ранжирования: ранг 1

присваивается самому высокому риску, ранг 4 – самому низкому риску.

2.2. АНАЛИЗ ВАРИАНТОВ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ МЕТОДОМ ПОРОГОВОГО АГРЕГИРОВАНИЯ

Множество проектов информационных систем с сервис-ориентированной архитектурой оценивается по группам критериев. Подобные задачи принятия решения, содержащие множество критериев и возможных альтернатив решения, называют многокритериальными задачами [11]. В подобных задачах наиболее часто применяется метод линейной свертки, а также методы принятия решения с учетом относительной важности критериев [10]. Однако в ряде задач использование этих методов представляется неэффективным в силу «компенсаторного» характера критериев, т.е. возможности критериев при агрегировании компенсировать низкие оценки по одним критериям высокими оценками по другим. В отличие от упомянутых методов пороговый метод агрегирования [3, 16] основан на построении результирующего ранжирования по совокупности критериев, представленных четырьмя рангами. Агрегирование осуществляется с помощью порогового правила, которое позволяет построить бинарное отношение, определяющее предпочтения на множестве проектов.

В [12] описывается набор критериев для сервис-ориентированной архитектуры. Определенные группы критериев проекта, кроме рисков, ранжируются по четырех- или пятибалльной шкале. Группа рисков ранжируется сначала методом, описанным выше. А уже затем методом порогового агрегирования – по четырехбалльной шкале [16].

Формула преобразования ϕ из [16] для случая четырех рангов функции агрегирования имеет вид:

$$\phi(x) = \sum_{j=1}^2 C_{n-V_j(x)+3-j}^{4-j} + V_4(x) + 1 = \sum_{j=1}^2 \frac{(n - V_j(x) + 3 - j)!}{(4 - j)!(n - V_j(x))!} + V_4(x) + 1,$$

$$\phi(x) = \frac{(n - V_1(x) + 1)(n - V_1(x) + 2)}{6} + \frac{n - V_2(x) + 1}{2} + V_4(x) + 1,$$

где n – число критериев для сравнения альтернатив; $V_g(x)$ – число g -х оценок в записях векторов x или y ; $C_k^{k+1} = 0$ для всех $k \in [0, m-1]$, и $C_{-1}^0 = 1$.

Максимально возможное число классов эквивалентности этого слабого порядка при длине векторов, равной n , и при значении компонент вектора из $\{1, 2, 3, 4\}$ сводится к расчету числа неупорядоченных n выборов из 4 элементов с повторением:

$$\max \phi(x) = C_{n+3}^n = \frac{(n+3)!}{n!3!} = \frac{(n+1)(n+2)(n+3)}{6}.$$

А в случае пяти рангов функция агрегирования приобретает следующий вид:

$$\phi(x) = \sum_{j=1}^3 C_{n-V_j(x)+4-j}^{5-j} + V_5(x) + 1 = \sum_{j=1}^3 \frac{(n-V_j(x)+4-j)!}{(5-j)!(n-V_j(x))!} + V_5(x) + 1,$$

где n – число критериев для сравнения альтернатив; $V_g(x)$ – число g -х оценок в записях векторов x или y ; $C_k^{k+1} = 0$ для всех $k \in [0, m-1]$, и $C_{-1}^0 = 1$.

Максимальное число классов эквивалентности для пяти рангов приобретает вид

$$\max \phi(x) = C_{n+4}^n = \frac{(n+4)!}{n!4!} = \frac{(n+1)(n+2)(n+3)(n+4)}{24}.$$

Более подробно метод порогового агрегирования из [16] рассматривается в пункте 3.3.

В результате подход ранжирования проектов построения ИС с сервис-ориентированной архитектурой состоит из четырех этапов, приведенных на рис. 2.

Обследование	Первоначальное ранжирование	Повторное ранжирование	Результаты анализа
<ol style="list-style-type: none"> 1. Определение альтернатив 2. Анкетирование архитекторов решений <ul style="list-style-type: none"> • Сбор данных по архитектуре решений • Сбор статистики ошибок текущих систем 3. Оценка альтернатив по вектору критериев 	<ol style="list-style-type: none"> 1. Ранжирование рисков <ul style="list-style-type: none"> • Категоризация рисков по 5 группам • Ранжирование групп рисков максимальным методом 2. Ранжирование проектов по группам критериев <ul style="list-style-type: none"> • Внутренняя организация ИС • Организация изменений и политик • Готовность процессов и бизнес-сервисов • Минимизация операционных рисков 	<ol style="list-style-type: none"> 1. Перевод в ранги результатов первоначального ранжирования 2. Повторное ранжирование альтернатив методом порогового агрегирования 	<p>Выбор проекта с наибольшим рангом как наиболее эффективного проекта</p>

Рис. 2. Описание общего подхода

2.3. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РИСКОВ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ И ВЫБОР ПРОЕКТОВ В СРЕДЕ АНАЛИЗА

В данной работе построена модель для имитации работы информационной системы с сервис-ориентированной архитектурой в части возникновения ошибок в ее отдельных компонентах. При этом выполняется моделирование всего процесса обработки ошибок от подачи заявки на решение проблемы до ее разрешения (процедура моделирования рисков). На основе этих данных выполняется сбор статистики по тем группам рисков информационной системы с сервис-ориентированной архитектурой, по которым данная статистика не была получена в рамках практического применения предлагаемых в работе методов. С помощью имитационной модели можно будет сделать прогноз частоты возникновения рисков и, соответственно, оценок, которые используются в модели выбора наиболее эффективного проекта реализации сервис-ориентированной архитектуры информационной системы. Также выполняется генерация таблицы рангов на основе свободного определения числа приоритетов и критериев. Программная реализация предложенных методик выбора

эффективного проекта реализации информационных систем с СОА и анализа операционных рисков СОА выполнена в качестве надстройки для одной из достаточно распространенных программ имитационного моделирования PowerSim 8.0 (далее «среда моделирования»). Для ранжирования рисков использовался Microsoft Excel (далее «среда анализа»). Текущая версия построенной имитационной модели еще весьма ограничена.

Имитационный эксперимент выполняется согласно правилу (2). Собрана статистика по множеству типов рисков t_0, \dots, t_n , $t_n \in T$.

Следующие параметры являются входными для имитационной модели:

$N = \langle n_h, n_{IO}, n_f, n_s, n_d \rangle$ – имеющиеся данные по частоте появления каждого типа риска с разбивкой по четырем приоритетам;

d – данные по средней длительности устранения ошибки за момент времени θ_k (в имитационной модели – квартал);

w_{ij} – стоимость человеко-дня для i -й функциональности в зависимости от j -й роли и приоритета сообщения.

На основе данных входных параметров в рамках имитационного эксперимента подбираются такие показатели, как среднее время устранения ошибки и среднее время подачи заявления. Также для каждого типа риска подбирается процентное соотношение приоритетов. Так, например, на основе введенных рисков по данным нефтегазового предприятия приоритеты 1, 2, 3 и 4 соотносятся как 10%, 10%, 70%, и 10%.

В случае если статистика отсутствует, устанавливаются типы распределений, соответствующих каждому типу рисков. Эти распределения используются для генерации потока ошибок по соответствующим типам рисков. Для этого используются четыре параметра: среднее время устранения ошибки, среднее время подачи заявления, математическое ожидание и дисперсия. На основе этих параметров проводится имитация процесса устранения ошибок, т.е. генерируется поток полученных ошибок и поток устраненных ошибок.

Также проводится ранжирование множества ошибок. При этом разным типам рисков t_0, \dots, t_n соответствует последовательность оценок $r_0, \dots, r_m, m \in [1, 4]$, в соответствии с правилом (2). При генерации потока ошибок он разделяется по приоритетам в соответствии с введенным соотношением приоритетов, чтобы в дальнейшем использовать эти данные для ранжирования. Например, сгенерированный поток ошибок данных с разбивкой по приоритетам приведен в таблице 1.

Таким образом, моделируется процесс обработки ошибок информационных систем с сервис-ориентированной архитектурой. Динамика показателей полученных и исправленных ошибок приведена на рис. 3. Здесь отражены результаты моделирования на основе статистики данных и с учетом того, что среднее время исправления ошибки составляет 26 дней, среднее время подачи заявления – 20 дней.

Из проведенных имитационных экспериментов можно сделать вывод, что для минимизации рисков необходимо сокращать время подачи заявлений, время устранения ошибок и число заявлений, обрабатываемых в единицу времени. Однако необходимо также понимать, что чем меньше сроки на устранение ошибок, тем больше остается неразрешенных ошибок.

Таблица 1. Поток ошибок данных по приоритетам (разбивка по кварталам)

Время	Ошибки данных	Приоритет			
		Очень высокий	Высокий	Средний	Низкий
04.2006	5,25	0,06	0,00	2,87	1,00
06.2006	14,98	1,50	1,50	10,49	1,50
10.2006	0,01	0,00	0,00	0,00	0,00
01.2007	7,32	0,73	0,73	5,13	0,73
04.2007	35,39	3,54	3,54	24,78	3,54
06.2007	23,74	2,37	2,37	16,61	2,37
10.2007	8,63	0,86	0,86	6,04	0,86
01.2008	12,81	1,28	1,28	8,97	1,28
04.2008	6,12	0,61	0,61	4,28	0,61
06.2008	1,81	0,18	0,18	1,27	0,18
10.2008	15,64	1,56	1,56	10,95	1,56

Время	Ошибки данных	Приоритет			
		Очень высокий	Высокий	Средний	Низкий
01.2009	0,26	0,03	0,03	0,18	0,03
04.2009	41,79	4,18	4,18	29,25	4,18
06.2009	32,72	3,27	3,27	22,91	3,27
10.2009	3,08	0,31	0,31	2,15	0,31
01.2010	2,85	0,29	0,29	2,00	0,29
04.2010	1,32	0,13	0,13	0,93	0,13
06.2010	11,67	1,17	1,17	8,17	1,17
10.2010	55,63	5,56	5,56	38,94	5,56
Итого	275,77	27,57	27,57	193,05	27,57

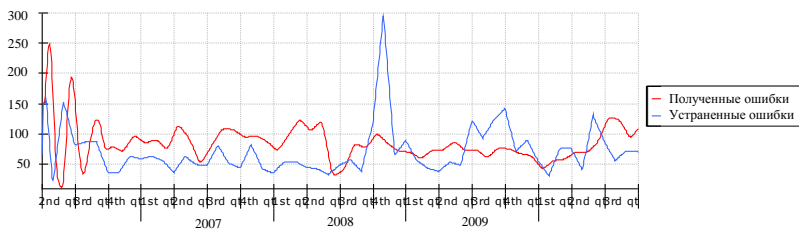


Рис. 3 Поток полученных и устраненных ошибок (вставка из среды Powersim Studio 8)

Таким образом, получаем оценки по всем типам рисков с разбивкой по приоритетам (таблица 2).

Таблица 2. Ранжирование рисков

Типы рисков	Приоритет				
	Очень высокий	Высокий	Средний	Низкий	Ранг
Ошибки ввода-вывода	15,43	15,43	108,00	15,43	2
Ошибки данных	27,577	27,577	193,039	27,577	2
Ошибки связующего ПО	7,72	7,72	54,06	7,72	4
Функциональные ошибки	51,14	51,14	358,01	51,14	1
Системные ошибки	28,31	28,31	198,15	28,31	2

Далее осуществляется перенос данных из среды моделирования в среду анализа, после чего эксперт запускает

процедуру генерации таблицы рангов. В соответствие с данной процедурой ранги генерируются автоматически с помощью разработанной автором специальной программы, использующей алгоритм ранжирования по пороговому правилу, приведенному выше.

В качестве примера приведем результаты генерации таблицы рангов для трех приоритетов и четырех критериев (таблица 3). Из приведенной таблицы видно, что, например, ранг 1 присваивается сочетанию оценок, в котором присутствуют три единицы. В соответствии с алгоритмом ранжирования такому сочетанию (или альтернативе) присваивается переходный ключ, равный трем, который преобразуется в ранг 1.

Таблица 3. Сгенерированные ранги

Ранг (ген.)	Ключ (ген.)	Сочетание числа повторений каждого приоритета (ген.)				Матрица сочетаний
		1	2	3	4	
1	3	3	0	0	0	(1, 1, 1)
2	6	2	1	0	0	(1, 1, 2)
3	9	1	2	0	0	(1, 2, 2)
4	12	0	3	0	0	(2, 2, 2)
5	18	2	0	1	0	(1, 1, 3)
6	21	1	1	1	0	(1, 2, 3)
7	24	0	2	1	0	(1, 2, 3)
8	33	1	0	2	0	(1, 3, 3)
9	36	0	1	2	0	(2, 3, 3)
10	48	0	0	3	0	(3, 3, 3)
11	66	2	0	0	1	(1, 1, 4)
12	69	1	1	0	1	(1, 2, 4)
13	72	0	2	0	1	(2, 2, 4)
14	81	1	0	1	1	(1, 3, 4)
15	84	0	1	1	1	(2, 3, 4)
16	96	0	0	2	1	(3, 3, 4)
17	129	1	0	0	2	(1, 4, 4)
18	132	0	1	0	2	(2, 4, 4)
19	144	0	0	1	2	(3, 4, 4)
20	192	0	0	0	3	(4, 4, 4)

Среда моделирования поддерживает графическое представление результатов (рис. 4), а также дает возможность загрузить документ для анализа в виде MS Excel. Результаты анализа также выгружаются в MS Excel. Они используются для дальнейшего анализа в интерфейсе, которым может быть порталное приложение или в случае прототипа – шаблон MS Excel с встроенными расчетами для применения многокритериального анализа методом порогового агрегирования, а также анализа рисков максиминным методом.

Целью создания инструмента является снижение трудоемкости процесса оценки рискованности сервис-ориентированной архитектуры на предприятии и поддержка принятия решений на основе полученного потока ошибок и ранжирования максиминным и многокритериальным анализом.

Развитие концепции инструмента предполагается на основе корпоративного хранилища данных SAP BW и порталного интерфейса.

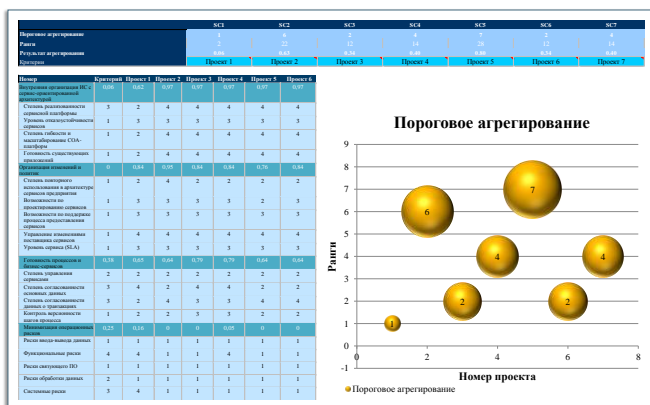


Рис. 4. Результаты ранжирования в инструменте

Пример использования предложенных методов анализа рисков и выбора оптимального проекта приведен более подробно в [12].

Сравним предложенный подход с существующими методами анализа и выбора проектов реализации информационных систем с использованием любого типа архитектуры, включая сервис-ориентированную архитектуру.

3. Известные модели выбора проектов реализации информационных систем с СОА и оценки рисков

Выбор проектов реализации информационной системы обосновывается либо эффективностью и успешностью внедрения, либо качеством системы и состоянием рынка ИС в данной области.

В силу отсутствия единого подхода к анализу рисков информационных систем, доказавшего свою эффективность и простоту использования, вопрос анализа рисков ИС в целом остается открытым. Еще более остро этот вопрос стоит при принятии решений о внедрении таких новых подходов как сервис-ориентированная архитектура. В литературе можно найти мало упоминаний о методах, показавших свою эффективность при оценке рискованности проекта с сервис-ориентированной архитектурой, поэтому обратимся к обзору более общей темы – анализу рисков ИС.

Каждый метод в виде первого шага определяет группы рисков. Одни подходы предлагают выявлять риски на основе влияния неудачи реализации СОА [23] на такие аспекты (и, соответственно, группы риска) как бизнес, планирование, стоимость и качество. Другие в данном случае ориентируются на этапы процесса разработки информационной системы [22], что дает три основные группы – риски проектирования продукта, риски среды разработки, и риски, возникающие из-за ограничений программы. В [18] для рисков уже используется более детальная классификация, включающая в себя аспекты персонала, ограничения бюджета и времени, вероятность разработки неверного функционала, меняющиеся потребности и даже внешние факторы влияния на проект.

Как объяснялось выше, риски сервис-ориентированной архитектуры тесно связаны с ресурсами или компонентами информационной системы. Для сравнения существующих

методов оценки рисков используем предложенную выше классификацию ресурсов ИС:

1. Человеческие ресурсы.
2. Прикладные ресурсы:
 - подсистема доступа;
 - прикладное ПО;
 - связующее ПО;
 - инфраструктурное ПО.
3. Технические ресурсы.

Ряд методов оценки рисков рассматривают эффект от информационной системы целиком, а некоторые подходы созданы для других областей. Поэтому добавим к классификации две группы: информационные системы в целом и применение не в ИС.

3.1. ОБЗОР МЕТОДОВ ОЦЕНКИ РИСКОВ ИНФОРМАЦИОННЫХ СИСТЕМ С СЕРВИС- ОРИЕНТИРОВАННОЙ АРХИТЕКТУРОЙ

Каждая модель оценки рисков из существующих сегодня поддерживает принятие решения с учетом определенных целей. Сравнение всех моделей оценки рисков приведено в таблице 4. Сравнение строится по степени охвата указанных типов ресурсов информационной системы, а также по возможности применения к ИС с сервис-ориентированной архитектурой в целом, ИС с другими типами архитектур (клиент-серверная, распределенная) и к другим областям, требующим оценки операционных рисков (т.е. моделям, несвязанным с информационными технологиями).

Для анализа используем пять рангов:

- – невозможность использования метода;
- ◐ – частичное использование для типа риска;
- ◑ – применимость метода в половине случаев при определенных условиях;
- ◒ – в целом метод применим за несколькими исключениями;
- – метод применим (таблица 4, см. в приложении).

Разделим модели оценки рисков на проактивные и реактивные. Проактивные модели ориентированы на определение и устранение рисков до их возникновения [22].

Реактивные модели предлагают расчет затрат или выполнение мер по устранению уже реализовавшихся рисков, или использование механизмов отслеживания приближения рисков.

Исходя из анализа, приведенного в таблице 4, видно, что предложенный автором данной работы метод является наиболее общим с точки зрения ресурсов ИС с сервис-ориентированной архитектурой по сравнению с остальными методами.

3.1.1. ЧЕЛОВЕЧЕСКИЕ РЕСУРСЫ

Данная категория используется для оценки рисков в большинстве существующих моделей. Все предложенные методы анализа рисков человеческих ресурсов (или рисков персонала) делятся на проактивные и реактивные.

Проактивные модели оценки рисков представлены в [4, 16, 30]. Они позволяют заблаговременно спланировать затраты при реализации рисков с помощью оценки вероятности риска и величины убытка соответственно [4, 30] либо на основе экспертной оценки индикаторов рисков [16], которые позволяют подсчитать суммарную величину убытков.

В [18] представлены техники проактивного и реактивного анализа рисков проекта. Функциональный метод анализа рисков проекта внедрения ИС утверждает, что влияние на проект оказывают 14 факторов (и, соответственно, категорий риска), среди которых степень распределенности модели данных, производительность системы, производительность конечных пользователей, уровень интерактивности обновления данных, легкость установки, сложность конфигурации, степень интерактивности ввода данных, сложность обработки данных, простота эксплуатации и степень поддержки изменений. К управлению человеческими ресурсами можно отнести только фактор производительности персонала. Степень влияния каждой из n -категорий выражается оценкой DI_n , которая принимает значения от 0 до 5. Общее влияние (VAF) в результате рассчитывается по формуле

$$VAF = 0,65 + 0,01 \sum_{n=1}^{14} DI_n.$$

где $DI \in [0; 5]$ – степень влияния.

Очевидно, что это лишь одна сторона и возможная область рисков при внедрении ИС. При этом подходе также не учитываются риски использования ИС, после того как она внедрена на предприятии, поэтому покрытие категории лишь на четверть.

Другим методом оценки риска, предложенным в [18], является подход Сосома II. При методе Сосома II определяется всего 6 групп проектных рисков, одна из которых – риски персонала. Суммарный риск (TR) рассчитывается как сумма уровней риска (RL) каждого типа рисков:

$$TR = \sum_{n=1}^6 RI_n .$$

Критерий риска для каждого модуля рассчитывается как $R = (TR/373) \cdot 100$. Определение уровня риска является экспертной оценкой. Недостатком данного подхода, кроме сложности оценки уровня риска, является и то, что не различаются разные приоритеты и влияния разных типов рисков.

Еще одна модель проектных рисков описана в [25]. Она предлагает реактивную модель, которая заключается в контроле индикаторов риска: изменчивость требований (RV), сложность (LGD) и эффективность (EF). Важно отметить, что в данной модели эффективность измеряется как отношение рабочего времени и времени нерабочего. Это является хорошим индикатором возможного риска персонала, поскольку полностью определяет все риски персонала и не зависит от экспертных оценок, однако есть и другие типы рисков персонала, которые данный метод не учитывает.

В [22] скорее представлены основные категории рисков информационных систем и предложены методы предотвращения появления рисков. Рассматриваются три основных модели управления рисками: управление рисками команды (TRM), постоянное управление рисками (CRM) и управление рисками программного обеспечения (SRM). TRM рассматривает как раз риски персонала, этапы их определения и

устранения. Покрытие категории персонала здесь полное, однако не предлагается методов оценки рисков.

Модель оценки рисков информационных систем с сервис-ориентированной архитектурой, предложенная автором данной статьи, так же как и в [25] ориентирована на определение и расчет рисков без использования экспертных оценок. Однако предложенная модель предлагает использовать проактивный подход оценки рисков (т.е. определение возможных рисков до их появления). Преимуществом данного подхода является то, что риски охватывают наиболее полный объем возможных затрат, в отличие от методов в [18, 22, 23], поскольку не пытаются определить выборочные типы рисков. Еще одно достоинство предложенной модели в области человеческих ресурсов – возможность определения рисков независимо от фазы информационной системы (предпроектная оценка, проект внедрения ИС, продуктивная эксплуатация ИС). Для этого, так же как и в [2], используется типы моделей распределения вероятностей. Модели определены для каждого типа рисков. Правда, в отличие от [2], предлагается определять распределение на основе ошибок ИС предыдущих лет.

3.1.2. ПРИКЛАДНЫЕ РЕСУРСЫ

Оценка прикладных ресурсов состоит из оценок типов прикладных ресурсов. Поэтому рассмотрим модели, используемые для оценки компонентов прикладных ресурсов ИС.

3.1.2.1. ПОДСИСТЕМА ДОСТУПА

Оценка рисков подсистемы доступа или интерфейса редко встречается в моделях. Среди рассмотренных моделей они определены всего в трех.

В [30] определена модель анализа рисков при дизайне ИС. Категории риска, определенные в модели, включают два типа подсистем доступа – клиентский компьютер («толстый» клиент) и веб-интерфейс. Автор предлагает определить риски, затем проранжировать и разработать стратегию предотвращения

рисков. Убыток же при реализации рисков оценивается с помощью следующего показателя:

$$ALE = SLE \times ARO,$$

где ALE – ежегодный ожидаемый убыток; SLE – единовременный ожидаемый убыток; ARO – годовая норма возникновения риска.

Подход дает общую оценку убытка от рисков, без выделения какой-либо группы или использования наиболее подходящей модели для каждой группы рисков. Ранжирование происходит с помощью линейной свертки, что имеет компенсаторный характер итогового ранга.

Оценка рисков подсистемы доступа частично приведена в [18]. В данном случае функциональный подход анализа рисков учитывает один из типов рисков подсистемы доступа – ввод данных через веб-интерфейс, снижающий безопасность данных.

В [22] определена категория рисков, связанных с пользовательским интерфейсом. Однако не приведен подход анализа рисков.

Модель оценки операционных рисков информационных систем с сервис-ориентированной архитектурой предлагает специализированный подход анализа операционных рисков подсистемы доступа в отличие от всех приведенных моделей. Оценка убытка на основе ошибок информационных систем предлагает комплексный подход анализа рисков этого типа.

3.1.1.2. ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

В [23] рассматривается один из типов рисков прикладного программного обеспечения (ПО) – архитектурная сложность, изменение комплекса информационных систем, отсутствие дизайна и недостаточность тестирования ИС. Для данных типов определяется вероятность возникновения риска, частота и стоимость убытка. Метод может наиболее точно определить убытка только при условии, что перечислены все типы возможных рисков.

Отдельно выделенная категория рисков приложений или прикладного ПО присутствует также в методах [18, 22, 30]. Эта категория в данных методах является одной из составляющих

общего ожидаемого убытка. В случае [30] она представлена в виде общей группы прикладного ПО, в [22] – в виде рисков дизайна и разработки, а в [18] – в виде двух из 14 характеристик системы (комплексная обработка, уровень транзакций) или категории процессные риски в методе Сосото II. Таким образом, оценка анализа данной категории рисков отдельно не проводится, но тип риска влияет на определение общей величины убытка.

Элементы рисков прикладного ПО рассматриваются и в модели [25] вместе с показателем сложности (*LGD*). Она измеряется на основе формальной спецификации как $LGD = O + D + T$, где *O* – число атомарных операций; *D* – число атомарных потоков данных; *T* – число абстрактных типов данных, необходимых системе. Этот показатель затем используется как параметр распределения Вейбулла, имитирующего проект разработки прикладного ПО.

Модель оценки операционных рисков информационных систем с сервис-ориентированной архитектурой, предложенная автором данной статьи, предлагает способ оценить риски прикладного ПО на основе статистики предыдущих лет. Это позволяет сделать подход и специализированным, и общим одновременно, поскольку с одной стороны отражает риски именно в области прикладного ПО в СОА, а с другой стороны позволяет дать наиболее точную оценку потенциальных рисков. В отличие от этого существующие методы дают оценку рисков прикладного ПО либо с помощью узко специализированных методов анализа, либо с помощью более общих методов, в которых риски прикладного ПО являются одной из категорий, влияющих на общий результат.

3.1.2.3. СВЯЗУЮЩЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Связующее ПО как категория рисков встречается еще реже, чем предыдущие категории. И объясняется это тем, что эффект интеграционной платформы обычно косвенный, поэтому оценку данной категории дать сложно. Один из немногих вариантов приведен в [18] в методе Сосото II, где напрямую определена группа рисков платформы. Однако не сказано, каким образом

определить события риска, относящиеся к данной группе. Другой пример приведен в методе [22]. Там определена группа рисков интеграции и тестирования и предложена модель CRM для устранения риска.

Уникальность и сильная сторона модели оценки операционных рисков информационных систем с сервис-ориентированной архитектурой, по сравнению с существующими методами, в данном случае заключается в том, что определить ошибки интеграционной платформы (иными словами – связующего ПО) значительно легче. А это дает возможность определить модель распределения вероятности связующего ПО и соответственно убыток от реализации рисков этой категории.

3.1.2.4. ДАННЫЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Риски данных, так же как и предыдущий тип ресурсов, выделяются как категория риска редко. Группа рисков данных выделена в модели рисков при дизайне ИС [30] и в функциональной модели [18]. В первой модели рассматриваются скорее риски базы данных или сервера, на котором построена информационная система, в то время как в функциональной модели среди 14 характеристик ИС можно найти риски, связанные с характеристиками распределения данных и их обработкой. И если в первом случае определяется вероятность возникновения таких рисков, то во втором дается экспертная оценка.

В отличие от этих методов, предложенная модель оценки операционных рисков информационных систем с сервис-ориентированной архитектурой в случае с данными, так же как и в предыдущих типах ресурсов, предлагает специализированный подход оценки убытка от реализации рисков данных. При этом данный подход анализирует все или большую часть возможных рисков.

3.1.2.5. ИНФРАСТРУКТУРНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Ошибки инфраструктурного ПО выделены как группа рисков в [23], где описаны риски неправильной работы инфраструктурного программного обеспечения и сбой серверного оборудования. Эти типы рисков используются для построения деревьев рисков, корнями которых являются нежелательные расходы, неопределенные требования, низкое качество и неправильное планирование. В случае если в перечисленные категории попали все возможные риски, подход определяет полную сумму убытков. Кроме того, метод предлагает проводить анализ до реализации проекта и всецело опирается на экспертную оценку.

Функциональный анализ рисков [18] включает категорию производительности. Как известно, производительность системы зависит от нескольких условий, в том числе и от правильно подобранного сервера и инфраструктурного ПО. Такой же тип риска упоминается и в [22].

Преимущество предложенной модели оценки операционных рисков информационных систем с сервис-ориентированной архитектурой в случае с инфраструктурным ПО заключается в том, что статистика, являющаяся базой для анализа рисков, обычно включает большее разнообразие ошибок инфраструктурного ПО и позволяет более четко определить типы возможных рисков. А убыток определяется исходя из каждого типа рисков.

3.1.3. ТЕХНИЧЕСКИЕ РЕСУРСЫ

Технические ресурсы практически не встречаются при анализе рисков информационных систем, хотя именно от работы сервера, частоты канала сети во многом зависит успех сервис-ориентированных приложений, постоянно обращающихся по сети к сервисам других систем.

Упоминание технических ресурсов есть только в [23], где речь идет о таких рисках, как сбой работы инструментов и потеря технического оборудования.

При использовании же предложенной модели оценки операционных рисков информационных систем с сервис-ориентированной архитектурой определяются и другие ошибки технических ресурсов, хотя ошибки ИС не всегда отражают все сбои в техническом оборудовании. Например, ошибка может говорить о временной недоступности системы, в то время как причиной ошибки может быть пожар в серверной комнате.

3.1.4. ИС С СОА

Данная группа рисков демонстрирует, что все существующие методы не были применены к оценке рисков специализировано для сервис-ориентированной архитектуры, кроме разработанного метода.

3.1.5. ИС В ЦЕЛОМ

Из таблицы 4 видно, что все рассмотренные методы, кроме [2, 19, 20], ориентированы на анализ рисков информационных систем независимо от типа архитектуры, т.е. все методы являются своего рода универсальными. Это плохо для таких архитектур информационных систем, как СОА, потому что риски возникают не только в стандартных подсистемах, но уникальных подсистемах, отличающих данный тип архитектуры.

Интересный вывод сделан в [25]: исследование и имитация нескольких проектов внедрения показали, что статистика проекта внедрения ИС имеет распределение Weibull.

Также есть ряд моделей, не связанных с типами ресурсов ИС. Так, например, при методе Coombs [18] определяются три характеристики проектных рисков: вероятность возникновения, сложность устранения и влияние на проект, если риск реализуется. Дополнительно определяется максимальное количество человеко-дней на устранение риска и максимальная стоимость устранения риска. Каждой характеристике присваивается вес (низкий – 1, средний – 3, высокий – 7). Сумма весов рисков показывает общий вес:

$$OverallWeight = \sum_{c=1}^3 W_c ,$$

где W_c – вес c -й характеристики.

Максимальная величина этого показателя будет равна 21 и это соответствует максимальной вероятности риска – 75%. Реальная частота появления риска вычисляется по формуле

$$C = \frac{OverallWeight}{OverallWeight_{max} * 0,75}.$$

Чтобы определить величину убытков, реальная частота появления умножается на максимальное количество человеко-дней на устранение риска и максимальную стоимость персонала. Преимуществами метода являются отображение влияния риска и возможность оценить стоимость риска с помощью оценки времени его устранения. Недостатком же является то, что обычно стоимость зависит не только от времени, необходимого для устранения риска, но и от приоритетности риска.

Этот недостаток устранен в модели оценки операционных рисков информационных систем с сервис-ориентированной архитектурой, предлагаемой автором.

3.1.6. ПРИМЕНЕНИЕ НЕ В ИТ

Ряд математических моделей, предложенных в [2, 19, 20], не связан с областью информационных систем. И все же заслуживает внимания, поскольку описывает наиболее популярные методы оценки рисков в других областях. Например, методология Basel II используется международными банками для того, чтобы заложить бюджет на предотвращение различных типов рисков, таких как рыночные, кредитные и операционные риски. По методологии Basel II банки часто измеряют риски, ассоциируемые с портфелем проектов [1]. Методология Basel II [1, 19] включает две методики анализа операционных рисков: стандартную и расширенную методику измерения (AMA). В первой методике общая прибыль разделяется на 8 линий бизнеса и умножается на фиксированную ставку процента, определенную по методологии Basel II. В соответствии с расширенной методикой банк должен иметь систему измерения внутренних

операционных рисков. Уровень риска определяется с учетом коэффициентов, отражающих вероятность наступления риска и его возможный размер. В [20] описываются модели операционных рисков на основе пуассоновского, биномиального и нормального распределений. Интересно, что в данной статье разные типы рисков имеют разные модели распределения:

1. Биномиальное распределение используется, чтобы описать распределение рисков, несущих значительный убыток для банка. В данном распределении вероятность наступления риска равна

$$P(n_k^{(b)} = n) = C_{N_k}^n p^n q_k^{N_k - n}, n \in \{0, \dots, N_k\}.$$

При этом $n_k^{(b)}$ – случайная величина, представляющая количество событий риска k -й группы; $C_{N_k}^n$ – число сочетаний из n по N_k ; p – вероятность того, что произойдет событие риска в каждый из n промежутков; q – вероятность того, что событие риска не произойдет; k – группа риска; N – число промежутков продолжительность τ каждый. Среднее значение величины убытка определяется по формуле $\langle Y_k^{(b)} \rangle = \langle n_k^{(b)} \rangle V_k^{(b)} = N_k p_k V_k^{(b)}$, $V_k^{(b)}$ – величина убытка от одного события риска k -й группы.

2. Пуассоновское распределение описывает события риска, которые возникают достаточно часто и требуют осуществления мероприятий, уменьшающих вероятность наступления риска. Вероятность наступления риска в данном распределении будет равняться

$$P(n_k^{(b)} = n) = \frac{(\mu_k)^n}{n!} e^{-\mu_k},$$

где μ_k – стационарное значение вероятности события риска данной группы.

Среднее значение величины убытка равно

$$\langle Y_k^{(p)} \rangle = \langle n_k^{(p)} \rangle V_k = \mu_k V_k^{(b)}.$$

3. Постоянно возникающие события риска, не приносящие таких больших убытков как предыдущие два типа, имеют

нормальное распределение. Плотность распределения в таком случае равняется

$$P(n) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(n-a)^2}{2\sigma^2}},$$

при этом среднее значение $\langle n \rangle = a$, а дисперсия $Dn = \sigma^2$.

Если говорить о предложенной модели оценки операционных рисков информационных систем с сервис-ориентированной архитектурой, ее можно применить как к информационным системам с СОА, так и с любой другой архитектурой. Однако для ее применения к другим областям деятельности, не связанным с ИТ, модель потребует замены статистической базы на другой источник статистических данных.

Стоимость убытков от информационных систем является хорошим показателем успешности проекта внедрения. Однако при принятии решения о переходе на сервис-ориентированную архитектуру и выборе правильного проекта реализации необходимо руководствоваться не только показателями рискованности проекта, но другими критериями успешности архитектуры СОА. Совокупность всех характеристик архитектуры и рисков ставит непростую задачу выбора проекта для лица, принимающего решение (ЛПР). Поэтому рассмотрим существующие методы выбора проектов.

3.2. МНОГОКРИТЕРИАЛЬНЫЙ АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ С СЕРВИС- ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ

Информационная система имеет несколько фаз жизненного цикла не зависимо от типа поддерживаемой архитектуры. В общем случае фазы ИС представлены на рис. 6.

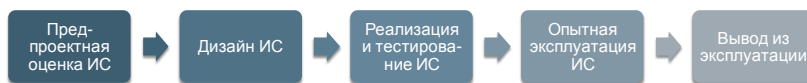


Рис. 5. Жизненный цикл ИС

Методы анализа и сравнения информационных систем применяются на разных стадиях жизненного цикла. Ряд публикаций ориентируется на выбор методологии внедрения СОА, оптимизирующей выгоды от архитектуры [4], которые рассматриваются обычно на фазе предпроектной оценки ИС. Также можно найти и описание подходов измерения успешности внедрения, т.е. анализ информационных систем на фазе опытной эксплуатации. Устоявшегося подхода сравнения проектов сервис-ориентированной архитектуры информационной системы на сегодняшний день не существует. И в научных журналах можно найти немного предложений по анализу таких проектов. Поэтому проведем более расширенный обзор существующих методов многокритериального анализа проектов информационных систем (как с сервис-ориентированной архитектурой, так и с другим типом архитектуры).

Метод выбора оптимального проекта СОА, предложенный автором в данной работе, универсален и может быть применен к ИС, находящейся на любой стадии жизненного цикла (таблица 5, см. в приложении), в том числе и в момент реализации ИС (т.е. на фазе реализации и тестирования ИС). В отличие от остальных рассмотренных методов, которые применимы либо до начала проекта реализации ИС с СОА [8, 14, 15, 21, 24, 26, 27], либо сравнивающих качество и функцию производительности уже реализованной ИС с СОА [14, 15, 17, 21, 22, 24, 27, 28, 31].

3.2.1. ПРЕДПРОЕКТНАЯ ОЦЕНКА ИС

В соответствии со стандартом [5] фаза предпроектной оценки ориентирована на определение потребностей пользователей в реализации ИС, исследовании и описании основных концепций ИС, а также их демонстрации и аттестации с целью уточнения характеристик системы, сравнения альтернативных вариантов реализации ИС.

Большинство методов сравнения и анализа эффективности поддерживают именно этот этап жизненного цикла ИС. Методы сравнения эффективности ИС предлагают принимать решение о

внедрении информационной системы независимо от того, какая архитектура внедряется на основании таких показателей (или критериев) как дисконтированный срок окупаемости инвестиций, чистый дисконтированный доход (*NPV*), индекс рентабельности инвестиций и внутренняя норма доходности (*IRR*).

В таблице 5 приведено сравнение всех рассмотренных методов по покрытию ими фаз жизненного цикла ИС. Из шести рассмотренных методов четыре ориентированы на данную фазу ИС.

В [8] предлагается использовать методологию системного анализа иерархических структур. Она основана на предположении о наличии у анализируемых альтернатив информационных систем множества общих критериев $S = \{S_1, S_2, \dots, S_m\}$, а ЛПР должно определить оптимальный набор критериев, способствующих достижению цели. Решением этой задачи является парное сравнение альтернатив на «интенсивность проявления критериев» и парное сравнение важности весов. Каждый объект оценивается по 9-балльной шкале. В результате строятся матрицы для каждого критерия, иллюстрирующие парное сравнение объектов на рассматриваемый критерий. Затем определяется максимальный собственный вектор, соответствующий максимальному собственному значению и проводится его нормализация. Сначала считаются веса объектов относительно критериев, затем относительно цели, и выбирается объект с наибольшим весом. При этом недостатком метода является невозможность достичь совершенной согласованности суждений (т.е. транзитивности и числовой согласованности), в силу неточности измерений на практике. Данный метод применяется для сравнения альтернатив проектов на фазе предпроектного анализа, когда принимается решение о будущей системе. Существенный недостаток этого метода еще и в том, что число критериев не определено заранее. Это усложняет процедуру сравнения на данной фазе, поскольку ЛПР на ранней стадии проекта не располагает подробными сведениями об архитектуре решения.

Ряд работ [14, 26, 27] посвящен оценке информационных систем на основе функции производительности. Они предлагают использовать подход экономической рациональности для оценки информационных систем с помощью ожидаемой полезности для лица, принимающего решение (ЛПР). Информационная система в данном случае сравнивается исходя из матрицы ценности (ценности информации).

Например, функция полезности [26] представляется как

$$U = U_{\pi\beta}(\eta, \alpha, \kappa) = B_{\pi\beta}(\eta, \alpha) - K_{\pi}(\kappa),$$

где $B_{\pi\beta}(\eta, \alpha) = \sum_{zya} \beta(a, z) \pi_z \eta_{zy} \alpha_{ya}$; $K_{\pi}(k) = \sum_z \pi_z \kappa_z$; x – набор входных данных ИС; y – набор исходящих данных ИС; a – действия или решения пользователя; η – функция трансформации данных; $y = \eta(x)$; π – некоторое распределение вероятностей; β – функция выгод; K – ожидаемая стоимость обработки данных; B – ожидаемые дисконтированные выгоды; $\{\alpha\}$ – набор всех трансформаций из y в a ; z – события, вышедшие из-под контроля пользователя.

Таким образом, ценность информации (V) в данном случае измеряется как

$$V_{\pi\beta}(\eta) = \max_{\alpha \in \{\alpha\}} B_{\pi\beta}(\eta, \alpha) = B_{\pi\beta}(\eta, \alpha^*)$$

где α^* – оптимальный набор.

Максимизация полезности в данном случае выглядит следующим образом:

$$\max_{\eta} U = \max_{\eta} V_{\pi\beta}(\eta) - \min_{\kappa} K_{\pi}(\kappa).$$

Этот метод прекрасно подходит для данной фазы ИС и ориентирован в основном на нее, поскольку использует по максимуму данные получаемые на этой фазе: план проекта и описание требований. Однако прогнозное значение функции полезности, конечно, может отличаться от реальных характеристик системы, полученных в ходе дальнейших фаз внедрения. Для комплексного анализа в таком случае необходимо применять систему других методов, позволяющих контролировать внедрение и использование ИС.

Другое направление [15, 21] предлагает использовать метод многокритериальной функции полезности для анализа проблем информационных систем, а затем выбрать систему, которая дает оптимальное решение функции. Автор [15] предлагает определить функцию полезности для пользователя(-лей) информационной системы. Предлагаемый подход основан на многокритериальной теории полезности, который комбинирует приведенные значения выгод. Несмотря на то, что метод основан на анализе «ex ante», т.е. до внедрения системы, ряд показателей обычно получаются уже после внедрения, поэтому для их прогнозирования используются экспериментальные оценки и значения, полученные из опросных листов и интервью. Автор [15] рассматривает следующие показатели функции полезности для информационной системы:

- Своевременность включает в себя время отклика и частоту ответа системы. Частота ответа имеет следующую функцию полезности: $U_i(t) = a_1 \exp(-b_1 t)$, где $a_1, b_1 > 0$ и a_1 служит как калибровочный коэффициент, который преобразует значение функции в цифровое значение.

- Содержание включает похожесть и уровень агрегации. Похожесть измеряется так: $U_s(s) = -a_2 \exp(-b_2 s)$, где $a_2, b_2 > 0$ и a_2 служит как калибровочный коэффициент, который преобразует значение функции U_s в цифровое значение. S основано на вероятности ошибок первого и второго типа, описанных в статье, $s = 1 - P_1 - P_2$, где P_1 – это, например, вероятность что существующая запись данных не извлечена и P_2 – вероятность того, что извлеченная запись не та, которую хотели получить.

- Формат сложно оценить с помощью каких-либо показателей в численном виде, поэтому для этого используются оценки пользователей.

Предположим, стоимость растет линейно при увеличении s и снижении t : $c = -a_3 t + b_3 s + const$, $a_3, b_3 > 0$. Объединенная функция полезности тогда $u = a_1 \exp(-b_1 t) - a_2 \exp(-b_2 s) - (-a_3 t + b_3 s) + const$. Решение данной функции должно определить оптимальную систему для организации. В случае если невозможно определить объединенную функцию

полезности, пользователь может проранжировать показатели. А затем при условии четкого доминирования одного из показателей можно выбрать оптимальную систему.

Для применения данного метода к частному случаю информационной системы с сервис-ориентированной архитектурой необходимо уточнение параметров оценки и соответственно более детальная проработка функции полезности с учетом предложенных параметров. Однако, как и в ряде предыдущих методов, оценка некоторых параметров количественно может оказаться затруднительной.

Метод оценки качественных параметров, помогающих ЛПР решать многокритериальные задачи, представлен в [24]. Здесь функция полезности определена для множества показателей следующим образом:

$$V(x) = \sum_{i=1}^n w_i v_i(x_i),$$

где n – количество показателей; i – число показателей; x_i – i -й показатель; v_i – i -я функция полезности от одного показателя (SAVF); w_i – вес i -го показателя; X – набор всех показателей; V – функция полезности от множества показателей (MAVF).

Набор показателей X не зависит от набора показателей Y . Для определения оптимального значения используется теория предпочтений. Доминирование устанавливается на основании серии вопросов, сравнивающих несколько параметров по нескольким альтернативным проектам. Далее параметры раскладываются в виде дерева анализа проблемы и матрицы принятия решения. Каждый узел дерева представляет значение атрибута, а связи – зависимости между атрибутами. Опрос помогает установить, какие предпочтения соответствуют какими вариантам. В результате часть альтернативных проектов отсеивается во время опроса. И последний шаг заключается в исключении доминируемых альтернатив. Таким образом, данный подход является последовательным, итеративным методом анализа альтернативных проектов с участием ЛПР, который помогает исключить неподходящие варианты. Недостатками

данного подхода являются неопределенность набора показателей и необходимость их определения для каждого проекта, а также необходимость участия ЛПР на протяжении практически всего анализа.

Все перечисленные проблемы устранены в предложенном методе многокритериального анализа. Он предлагает четко определенную классификацию характеристик ИС с сервис-ориентированной архитектурой. Полученные варианты реализации ранжируются с помощью многокритериального анализа методом порогового агрегирования. При этом на фазе предпроектного анализа метод позволяет проанализировать альтернативы с учетом имеющихся на этот момент деталей архитектуры.

3.2.2. ДИЗАЙН ИС

В соответствии со стандартом [5] данная фаза жизненного цикла ИС охватывает «период проектирования, создания, интеграции (сборки), тестирования и оценки системных технических средств, компьютеров, программных средств, оборудования, персонала подсистем, его обучения и объектов сопровождения».

На этом этапе часто приходится принимать решение о выборе более подходящего проекта реализации информационной системы под конкретную группу процессов, поскольку выбор по определенным причинам не был сделан на ранней фазе, поэтому выбор делается с появлением более детальной информации на этапе дизайна. Однако не так много существует методов, ориентированных на поддержку этого выбора. Один из методов, применяемых на данном этапе – анализ альтернатив проектов методом системного анализа иерархических структур [8]. Уязвимость метода, как и раньше, в неопределенности критериев принятия решения, это может привести к одностороннему анализу проекта.

В отличие от этого метода, идея предложенного метода выбора оптимального проекта СОА заключается в том, чтобы исключить приведенный недостаток за счет идентификации критериев. Критерии СОА дают возможность сравнения

альтернатив как на начальном этапе проекта, так и при дизайне ИС. С другой стороны, это делает метод четко специализированным под требования сервис-ориентированной архитектуры.

3.2.3. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ИС

В соответствии с [5] во время данной фазы спроектированную и разработанную систему адаптируют под конкретных пользователей, копируют систему на носители или аппаратные средства, готовые для промышленного использования, готовят сопроводительные документы и проводят окончательное тестирования ИС. Целями данной фазы являются «квалифицированное изготовление и поставка работоспособной и сопровождаемой системы заказчику».

Фаза реализации и тестирования обычно уже не предполагает, что решение будет заменено, поскольку ориентирована на внедрение выбранного решения. Однако в некоторых случаях данная фаза требует поддержки принятия решения при реализации дополнительных или неочевидных ранее требований. В таких случаях обычно принимается быстрое решение в зависимости от приоритетности задач. Когда возникает такая необходимость, определяется стратегическое направление подсистемы ИС, что также требует сравнения альтернативных вариантов. Как и на предыдущей фазе, это можно сделать с помощью методов системного анализа иерархических структур [8], а в случае сервисного подхода – предложенным методом выбора оптимального проекта.

В качестве проверки правильности дизайна и реализации информационной системы предлагается применять метод оценки производительности ИС. В [17] описываются методы оценки производительности информационной системы безотносительно к типу архитектуры, и предлагается подход, основанный на многокритериальном анализе дизайна информационной системы. Информационная система оценивается с точки зрения системной и пользовательской. Системная точка зрения ориентирована на показатели использования ресурсов, стоимости и эффективности. С

пользовательской точки зрения больше интересны показатели пропускной способности, надежности и времени отклика. Подход предлагает три этапа анализа: оценка системы (определяет веса по критериям для анализируемой информационной системы), оценка целей пользователей (анализирует степень достижения целей заказчика с помощью метода многоцелевого программирования (MGP)) и оценка дизайна (подытоживает общую оценку информационной системы с учетом предыдущих двух). При этом запрос пользователя характеризуется набором операций A_1, A_2, \dots, A_N . Для измерения меры достижения цели для пользователя предлагается использовать разницу между ожидаемыми целями и достигнутой производительностью системы:

$$D(i) = G(i) - \sum_{j=1}^N R_j(i),$$

Где i – число целей; j – число операций; $R_j(i)$ – оценка средней производительности шага; $G(i)$ – целевая производительность.

В результате основная формула для оценки информационной системы выглядит следующим образом:

Максимизировать $\lambda = F^+ P^+ D^+ + F^- P^- D^-$ при условии, что $R\beta + D^- - D^+ = G$, где R – матрица текущего уровня производительности; β – матрица переменной принятия решения; D^+, D^- – матрица перевыполнений и невыполнений; G – матрица целевых производительностей шагов; F, P – матрицы критериев приоритетности и весов.

Решив уравнение, получим матрицу β – производительности, необходимой, чтобы минимизировать разницу целей и достичь цели пользователя. Из-за отличий целей пользователей от конечных целей внедряемой информационной системы внедрение часто заканчивается провалом. Однако этот подход ориентирован на то, чтобы сгладить этот недостаток и предложить удовлетворительный вариант системы. Также интересно, что в данном случае каждая альтернатива информационной системы анализируется как вариант архитектуры ИС, основанной на пользовательских шагах. Каждый пользователь осуществляет разные действия и,

соответственно, использует разные сервисы и элементы аппаратного оборудования.

Недостатки метода заключаются в том, что на практике довольно сложно оценить численно все цели. А если говорить о сервис-ориентированной архитектуре, то ее основная цель – гибкость ландшафта и скорость адаптации новых требований. Данные цели довольно тяжело перевести в цифровые значения, а влияние на численные показатели лишь косвенное. К тому же еще одним недостатком является предположение о линейных зависимостях последовательности операций. В случае нелинейной зависимости или параллельно протекающих шагах метод необходимо усовершенствовать и проработать более детальное сравнение.

В отличие от этого, в предложенном методе выбора оптимального проекта цели четко определены и отражены в критериях сравнения архитектуры. Однако специфика предлагаемого метода ограничивает его применение для данной фазы.

3.2.4. ОПЫТНАЯ ЭКСПЛУАТАЦИЯ ИС

По [5] данная фаза включает «эксплуатацию, применение и использование системы пользователями».

Когда система уже используется, можно оценить насколько она соответствует требованиям пользователей и их ожиданиям. Интересно, что из шести рассмотренных методов пять можно применить на данном этапе. Часть из них специализируется на данной фазе, другие же пересекаются с методами, используемыми на этапе предпроектного анализа.

Ряд работ [28, 31] посвящен оценке качества услуг, предоставляемой информационной системы независимо от типа архитектуры. Качество услуг основано на сравнении того, что клиент/пользователь хочет получить от ИС, и того, что реально предложено. В [31] предлагается оценить качество услуг с точки зрения пяти измерений: материального (средства обслуживания, оборудование и персонал), надежности (возможность предоставлять сервис точно и в срок), ответственности (желание помочь клиенту и предоставить сервис), гарантированности

(знания и вежливость сотрудников и их возможность вызывать доверие и уверенность), симпатии (заботливое, индивидуальное внимание предоставляемое клиенту). Таким образом, качество услуг измеряется в соответствии с формулой $G = P - E$, где G – результат оценки потребления сервиса; P – восприятие; E – ожидания. Оценки собираются в виде опроса. Сравнение проводится несколько раз, в зависимости от цикла внедрения информационной системы.

Измерение качества услуг информационной системы, таким образом, используется как еще одна составляющая при оценке эффективности ИС. При этом объектом измерения результата может быть как конкретная информационная система, так и ИТ-отдел в целом, исходя из всего комплекса информационных систем, внедренного и поддерживаемого департаментом.

Все методы, ориентированные на анализ ИС на фазе предпроектной, применимы и в данном случае: метод оценки производительности ИС [17], метод оценки функции полезности [14, 26, 27], метод оценки функции полезности проблем ИС [15, 21] и многокритериальная функция полезности [24]. Все эти методы рассматривают возможности смены системы в случае необходимости.

Предложенный метод выбора оптимального проекта СОА позволяет в данном случае сосредоточиться на специализированном типе систем и провести сравнение альтернатив в этой области. Преимуществом данного метода, по сравнению с остальными, является рассмотрение критериев ИС. Это позволяет определить наиболее уязвимую область рассматриваемых альтернатив, и выбрать оптимальный проект по набору критериев. В отличие от предлагаемых подходов, которые ориентируются на более обобщенный показатель эффективности.

3.2.5. ВЫВОД ИЗ ЭКСПЛУАТАЦИИ ИС

По [5] данная фаза используется для снятия системы из эксплуатации, архивирования данных, закрытия доступа к вводу данных в систему и ограничение поддержки ее пользователей.

Этап вывода системы из эксплуатации обычно предполагает замену ИС, перевод всех необходимых данных на новую платформу и определение стратегии развития класса систем. В общем случае, на этом этапе решение уже принято. Однако практика показывает, что решение об общем направлении ИС не всегда учитывает все детали перехода и нюансы покрытия требований новой системой. В частности может оказаться, что выбранная система не покрывает каких-либо требований. В таком случае необходимо определить дополнительную подсистему, возможно на базе другой платформы. Таким образом, методы оценки производительности ИС [17], методы оценки функции полезности ИС [14, 24, 26, 27] применимы и здесь. Как в предыдущих фазах жизненного цикла ИС, чем детальнее проводится сравнение, тем лучше выбранная стратегия. А уровень определения подсистемы ИС и работы с несколькими дополнительными требованиями сильно сужает действие существующих методов, в отличие от предложенного нового метода выбора оптимального проекта ИС с сервис-ориентированной архитектурой.

3.3. МЕТОДЫ РАНЖИРОВАНИЯ И ПОРОГОВОГО АГРЕГИРОВАНИЯ

Получив большое количество разных параметров оценки архитектуры, необходимо вывести итоговое решение. Обычно для агрегирования результатов используется линейная свертка, как видно из приведенных выше методов. Поскольку в области информационных систем на сегодняшний день не применяются другие методы агрегирования для принятия решений при выборе проектов реализации ИС, обратимся к методам ранжирования и порогового агрегирования, применяемым в других областях.

В основе ранжирования по методу порогового агрегирования [16] лежит «пороговое правило»

$Wtr = \{(x, y) | [v_1(x) < v_1(y)] \text{ или } [v_1(x) = v_1(y) \text{ и } v_2(x) < v_2(y)] \text{ или } [v_1(x) = v_1(y) \text{ и } v_2(x) = v_2(y) \text{ и } v_3(x) < v_3(y)]\}$, где $v_1(x)$ – число единиц («1») в записи вектора x , $v_2(x)$ – соответственно, число

двоек («2»), а $v_3(x)$ – число троек («3»). Таким образом, отношение Wtr представляет собой множество пар векторов, для которых либо в первом сравниваемом векторе число единиц меньше, чем во втором, либо при равном числе единиц число двоек в первом векторе меньше, чем во втором. Результатом агрегирования является ранжирование векторов.

Исходя из метода [16], оценки критериев делаются по шкале $\{1, 2, \dots, m\}$, где $m \geq 3$. Функция φ считается многокритериальным пороговым правилом, если она ранжирует два n -критериальных вектора x и y следующим образом:

- пусть g – самый маленький ранг, который попадает в векторах x и y ;
- если ранг g попадает более часто в векторе x , тогда $\varphi(x) < \varphi(y)$;
- если ранг g попадает более часто в векторе y , тогда $\varphi(x) > \varphi(y)$;
- если ранг g вообще не попадает в векторах (в силу того, что все ранги одинаковы в векторах x и y), тогда $\varphi(x) = \varphi(y)$.

При этом функция φ вычисляется по формуле

$$\Phi(x) = \sum_{j=1}^m C_{n-v_j(x)+m-j-1}^{m-j} \text{ для всех } x \in X,$$

такое что $C_k^{k+1} = 0$ для всех $k \in [0; m-1]$ и $C_{-1}^0 = 1$,

где $v_j(x)$ – число рангов j в записи вектора x .

Тогда действуют следующие аксиомы:

- *Парето-доминирование*: если координаты вектора x не меньше координат вектора y , и есть хотя бы одна координата в x строго больше соответствующей координаты y , то $\varphi(x) > \varphi(y)$.
- *Попарная компенсируемость критериев*: если все координаты векторов x и y равны, кроме некоторых двух, в которых значения x и y «взаимно обратные», то $\varphi(x) = \varphi(y)$.
- *Пороговая некомпенсируемость*: для любого ранга g такого, что $1 \leq g \leq m$: $\varphi(g-1, \dots, g) < \varphi(g, \dots, g)$. Например, если хотя бы одна координата в векторе x равна 1, то $\varphi(1, \dots, n) < \varphi(2, \dots, 2)$.

• *Аксиома редукции*: Если в двух векторах x и y значения по одной из координат равны, то эту координату можно не учитывать в решении вопроса о взаимном предпочтении этих векторов.

$$\forall x, y \in A, \exists i \ x_i = y_i \Rightarrow$$

$$\Rightarrow \varphi_n(x) > \varphi_n(y) \Leftrightarrow \varphi_{n-1}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) > \varphi_{n-1}(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n).$$

4. Заключение

Предложенный в данной статье метод позволяет осуществить выбор проекта реализации ИС с сервис-ориентированной архитектурой, который зависит от оценок по совокупности критериев в четырех основных областях: внутренняя организация ИС с сервис-ориентированной архитектурой, организация изменений и политик, готовность процессов и бизнес-сервисов и минимизация операционных рисков. Существует большое количество разнообразных методов анализа рисков, сравнения проектов ИС и определения их эффектов. Однако не каждый метод применим без изменений к случаю СОА. В статье приведено подробное сравнение существующих методов анализа рисков ИС. Сравнение осуществлялось на основе покрытия классов рисков каждым методом. Похожий подход сравнения применен и к методам выбора проектов реализации ИС, только в данном случае проверялось покрытие различных фаз жизненного цикла ИС. С учетом достоинств и недостатков рассмотренных методов был разработан метод анализа рисков сервис-ориентированной архитектуры, предлагаемый автором в данной статье.

В качестве применения предложенного подхода приведено описание созданного прототипа инструмента. Он используется не только для имитации потока рисков при работе ИС с сервис-ориентированной архитектурой в зависимости от типа рисков, но и имеет подсистему анализа со встроенной функцией порогового агрегирования для поддержки выбора оптимального проекта.

Благодарность

Автор благодарен профессору Ф.Т. Алескерову за постановку задачи.

Литература

1. АЛЕСКЕРОВ Ф.Т., АНДРИЕВСКАЯ И.К., ПЕНИКАС Г.И., СОЛОДКОВ В.М. *Анализ математических моделей Базель II*. – М.: ФИЗМАТЛИТ, 2010. – С. 288
2. АЛЕСКЕРОВ Ф.Т., БЫКОВ А.А., КУРМАКАЕВА Е.А. *Анализ операционных убытков на основе биномиального, пуассоновского и нормального распределений // Банковское дело*. – 2011. – №6. – С. 52–57.
3. АЛЕСКЕРОВ Ф.Т., ЯКУБА В.И. *Метод порогового агрегирования трехградационных ранжировок // Доклады Академии наук*. – 2007. – Т. 413, №2. – С. 181–183.
4. БИБЕРШТЕЙН Н., БОУЗ С., ДЖОНС К., ФИММАНТ М., ША Р. *Компас в мире сервис-ориентированной архитектуры (COA): ценность для бизнеса, планирование и план развития предприятия / Пер. с англ.* — М.:КУДИЦ-ПРЕСС, 2007. – С. 256.
5. ГОСТ Р ИСО/МЭК 12207-2002. *Информационная технология. Процессы жизненного цикла программных средств. Руководство по применению*. ГОССТАНДАРТ РОССИИ. – Москва, 2001
6. ИЗБАЧКОВ Ю.С., ПЕТРОВ В.Н. *Информационные системы: Учебник для вузов*. 2-е издание. – СПб.: Питер, 2006. – 656 с.
7. КОГАЛОВСКИЙ М.Р. *Перспективные технологии информационных систем*. (Серия «ИТ-Экономика»). – М.: ДМК Пресс; Компания АйТи, 2003. – 288 с.
8. ЛОБАНОВА А.А., ЧУГУНОВА А.В. *Энциклопедия финансового менеджмента*. – М.: Альпина Бизнес Букс, 2007. – С. 878.

9. ЛОТОВ А.В., ПОСПЕЛОВА И.И. *Многокритериальные задачи принятия решений*: Учебное пособие. – М.: МАКС Пресс, 2008. – 197 с.
10. ПОДИНОВСКИЙ В.В. *Анализ задач многокритериального выбора методами теории важности критериев при помощи компьютерных систем поддержки принятия решений* // Известия РАН. Теория и системы управления. – 2008. – №2. – С. 64–68.
11. ПОДИНОВСКИЙ В.В., НОГИН В.Д. *Парето-оптимальные решения многокритериальных задач*. – М.: ФИЗМАТЛИТ, 2007. – С. 256
12. ПЫРЛИНА И.В. *Выбор эффективного проекта реализации сервисно-ориентированной архитектуры информационной системы* // Проблемы управления РАН. – 2012. – №4. – С. 59–68.
13. ПЫРЛИНА И.В. *Классификация операционных рисков при сервисно-ориентированном подходе к созданию информационной системы* // Бизнес-Информатика. – 2011. – №4(18). – С. 54–61.
14. ACKOFF R.L. *Towards a Behavioral Theory of Communication* // Management Science. – 1958. – Vol. 4, №8. – P. 218–234.
15. ANITUV N. *A Systematic Approach toward Assessing the Value of an Information System* // MIS Quarterly. – 1980. – Vol. 4, №4. – P. 61–75.
16. ALESKEROV F.T., CHISTYAKOV V.V., KALYAGIN V.A. *Multiple criteria threshold decision making algorithms*: Working paper WP7/2010/02. – Moscow: State University – Higher School of Economics. – 2010. – P. 40.
17. CHANDLER J.S. *A Multiple Criteria Approach for Evaluating Information Systems* // MIS Quarterly. – 1982. – Vol. 6, №1. – P. 61–74.
18. DASH R., DASH RAJ. *Risk Assessment Techniques for Software Development* // European Journal of Scientific Research, ISSN 1450-216X. – 2010. – Vol. 42, №4. – P. 629–636.

19. *Developments in Modeling Risk Aggregation* // Bank for International Settlements Communications CH-4002 Basel, Switzerland, ISBN 92-9197- 847-7. – 2009 – P. 32.
20. EMBRECHTS P., PUCETTI G. *Risk Aggregation. Copula Theory and its Applications* / Eds. P. Jaworski, F. Durante, W. Haerdle, T. Rychlik. Lecture Notes in Statistics / Proc. 198, Springer Berlin/Heidelberg. – 2010. – P. 111–126.
21. GROCHOW J.M. *A Utility Theoretic Approach to Evaluation of a Time-Sharing System* // Statistical Computer Performance Evaluation / In Freiburger W. – Academic Press, New York. – 1972. – P. 25–50.
22. HIGUERA R.P., HAIMES Y.Y. *Software Risk Management* // Technical Report CMU/SEI-96-TR-012 ESC-TR-96-012. – 1996. – 48 p.
23. HOODAT H., RASHIDI H. *Classification and Analysis of Risks in Software Engineering* // World Academy of Science, Engineering and Technology. – 2009. – №56. – P. 446–452.
24. KLEIN G., BECK P.O. *A Decision Aid for Selecting among Information System Alternatives* // MIS Quarterly. – 1987. – Vol. 11, №2. – P. 177–185. [Электронный ресурс]. – URL: <http://www.jstor.org/stable/249360> (дата обращения 03.09.2013).
25. LUQI J.N. *A Risk Assessment Model for Evolutionary Software Projects* // IEEE [Электронный ресурс]. – URL: <http://www.disi.unige.it/person/ReggioG/PROCEEDINGS/luqi.pdf> (дата обращения 04.09.2013).
26. MARSCHAK J. *Economics of Information Systems* // Journal of the American Statistical Association. – 1971. – Vol. 66, №333. – P. 192–219.
27. PITT L.F., WATSON R.T., KAVAN C.B. *Service Quality: A Measure of Information Systems Effectiveness* // MIS Quarterly. – 1995. – Vol. 19, №2. – P. 173–187.
28. SHOLLER D. *2008 SOA User Survey: Adoption Trends and Characteristics* // Gartner Research ID Number: G00161125. – 2008. – 17 p.

29. VERDON D., MCGRAW G. *Risk Analysis in Software Design* // IEEE computer society/IEEE security & privacy. – 2004. – Vol. 2, № 4. – P. 79–84.
30. WATSON R.T., PITT L.F., KAVAN C.B. *Measuring Information Systems Service Quality: Lessons from Two Longitudinal Case Studies* // MIS Quarterly. – 1998. – Vol. 22, №1. – P. 61–79.

SERVICE-ORIENTED ARCHITECTURE: OPERATIONAL RISKS AND CHOICE OF EFFICIENT SYSTEM

Irina Pyrlina, Higher School of Economics, Moscow, Specialist, IT Architecture Expert (Irina.pyrlina@gmail.com).

Abstract: We survey existing methods of efficient implementation projects selection for information system with service-oriented architecture (SOA), methods of risks assessment and multi-criteria analysis applied to information systems with SOA. We also suggest a new method to evaluate operational risks due to SOA and develop multi-criteria threshold algorithms to support ranking of information system implementation projects. Methods were used to design a tool consisting of two key subsystems – a simulation environment and a multi-criteria analysis module.

Keywords: multiple criteria threshold algorithm, service-oriented architecture, SOA operational risks, efficiency criteria for information system architecture, project ranking.

*Статья представлена к публикации
членом редакционной коллегии М.В. Губко*

*Поступила в редакцию 29.08.2012.
Опубликована 30.09.2013.*

ПРИЛОЖЕНИЕ.

Таблица 4. Сравнение методов оценки рисков

Типы ресурсов	Методы и модели анализа рисков ИС									
	Вероятностный анализ дерева рисков [23]	Риски при дизайне ИС [30]	Функциональный анализ рисков [18]	Socomo II [18]	Coombes et al [18]	Модель проектных рисков [25]	BASEL II [19], Агрегирование рисков [16]	Управление рисками ПО [22]	Анализ операционных убытков [2]	Оценка рисков COA
1. Человеческие ресурсы										
2. Прикладные ресурсы										
Подсистема доступа										
Прикладное ПО										
Связующее ПО										
Данные ИС										

Типы ресурсов	Методы и модели анализа рисков ИС									
	Вероятностный анализ дерева рисков [23]	Риски при дизайне ИС [30]	Функциональный анализ рисков [18]	Соотношение П [18]	Coomb's et all [18]	Модель проектных рисков [25]	BASEL II [19], Агрегирование рисков [16]	Управление рисками ПО [22]	Анализ операционных убытков [2]	Оценка рисков COA
Инфраструктурное ПО										
3. Технические ресурсы										
4. ИС COA с										
5. ИС в целом										
6. Применение не в ИТ										

Таблица 5. Сравнение методов анализа и выбора проектов реализации ИС

Фазы ИС	Методы анализа и сравнения проектов реализации ИС					
	Метод систем-ного анализа иерархич. структур [8]	Оценка производительности ИС [17]	Качество услуг [28, 31]	Функция полезности ИС [14, 26, 27]	Многокритери-альная функция полезности [15, 21, 24]	Метод выбора оптимального проекта COA
Предпроектная оценка						
Дизайн ИС						
Реализация и тестирование ИС						
Опытная эксплуатация ИС						
Вывод из эксплуатации ИС						