

КОРПОРАТИВНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ

Управление созданием бизнес-ценности в организационных ИТ-проектах

Business-Value Creation Management in Organizational IT-Projects

DOI: 10.12737/article_595f74fbfcdc7.02113920

Получено: 14.01.2017 г. / Одобрено: 10.02.2017 г. / Опубликовано: 30.06.2017 г.

Зуйков К.А.

Аспирант, Национальный исследовательский университет «Высшая школа экономики», Россия, 101000, г. Москва, ул. Мясницкая, д. 20, e-mail: kirill.zuykov@gmail.com

Zuykov K.A.

Graduate Student, National Research University Higher School of Economics, 20, Myasnitskaya St., Moscow, 101000, Russia, e-mail: kirill.zuykov@gmail.com

Аннотация

В статье описывается новый подход к управлению созданием ценности в проектах организационных изменений с использованием ИТ-систем (организационных ИТ-проектах), основанный на применении сервисной модели продукта, управлении по стадиям и мониторинге ценности. Подход решает проблему невозможности контроля бизнес-ценности, создаваемой в проекте и состоящей в приращении ценности (стоимости) бизнеса в результате использования продукта проекта по причине реализации этой ценности на этапе постпроекта. Исследуются проблемы контроля ценности при использовании традиционного жизненного цикла ИТ-проекта и предлагаются пути решения. Описывается метод определения метрик ценности на основе сервисной модели продукта проекта и метод реализованной ценности — метод мониторинга, основанный на методе освоенного объема и предполагающий его дополнение вектором индексов реализации ценности с анализом их значений совместно с индексом выполнения сроков. Метод использует изменения в содержании проекта как инструмент управления созданием бизнес-ценности. Контроль риска расползания требований осуществляется при помощи показателя запаса устойчивости, рассчитываемого по ретроспективным данным о ранее выполненных изменениях. Для обоснования формулы расчета запаса устойчивости в работе приводится вывод и верификация эмпирическими данными формулы возрастания трудозатрат на изменения в зависимости от времени.

Ключевые слова: управление изменениями, ИТ-проекты, устойчивость проекта, расползание требований, бизнес-ценность.

Abstract

The paper is devoted to the new approach to value creation management in IT-based organizational change projects that utilizes the product service model, management by stages and value control. The approach solves the problem that a business-value created in a project is out of control because it is realized in the post-project stage. The paper studies value control problems that are conditioned by the utilization of the traditional IT-project life-cycle. The method of value metrics definition that is based on the product service model is presented together with the released value method a new method of value control that extends the earned value method with a vector of value creation performance indices. Those are to be considered in conjunction with the schedule performance index. Scope changes are used as an instrument to control the value creation process. The scope creep risk control is performed by calculation of stability margin based on previously done changes data.

Keywords: change management, IT-projects, project stability, scope creep, business-value.

Введение

Статья посвящена проблеме создания бизнес-ценности в проектах организационных изменений, неотъемлемой частью содержания которых является создание или внедрение информационной системы. В качестве примера можно привести проекты автоматизации бизнес-процессов, внедрения ERP, CRM-систем, информационных систем управления проектами и др. Исследуемая проблема состоит в том, что выполнение проекта в полном объеме в установленные сроки с затратами, не превышающими приемлемые для заказчика, не гарантирует, что продукт проекта создаст ожидаемую ценность для бизнеса. Большинство наиболее распространенных стандартов управления проектами отмечает важность управления создаваемой ценностью в проектах и дает определение бизнес-ценности.

«Бизнес-ценность — концепция, уникальная для каждой организации. Бизнес-ценность определяется как вся ценность организации, общая сумма всех материальных и нематериальных элементов. Примерами материальных элементов являются денежные активы, основные средства, акционерный капитал и коммуникации. К примерам нематериальных элементов относятся репутация, узнаваемость марки, общественное благо и торговые марки. ... Ценность может быть создана путем эффективного управления операционной деятельностью. Однако благодаря результативному применению дисциплин управления проектом, программой и портфелем организации приобретают способность применять надежные признанные процессы для достижения стратегических целей и получения большей бизнес-ценности от своих инвестиций в проект» [22].

P2M определяет проект как «мероприятие по созданию ценности в соответствии с определенной миссией. Конкретная миссия проектов может быть определена как поставка конкретных ценностей определенным заинтересованным сторонам. Успешное завершение проекта означает, что ценность, на получение которой нацелен проект, была достигнута» [21].

Что касается ИТ-проектов, то их ценность для бизнеса определяется влиянием внедряемой технологии на эффективность деятельности организации, а также на реализацию ее стратегии [9; 26]. Вклад информационных технологий в создание бизнес-ценности состоит либо в реализации определенных рыночных возможностей посредством создания новых продуктов и услуг или улучшения старых, либо в оптимизации бизнес-процессов для получения обычного результата с меньшими затратами [19; 24].

То есть ценность, создаваемая в проектах организационных изменений, состоит в приращении ценности (стоимости) организации вследствие реализации проекта. Эта ценность создается в операционной деятельности организации как результат использования продукта проекта на стадии постпроекта, и в проекте бизнес-ценность непосредственно не контролируется. Необходимо искать доступные для контроля в проекте факторы, оказывающие влияние на ценность. Из имеющихся методов управления проектами ближе всего к контролю поставки ценности подходит метод *PRINCE2* с реализованной в нем концепцией выгод и ролью старшего пользователя в управляющем совете проекта, ответственной за поставку запланированных выгод [20]. При этом *PRINCE2* не включает никаких конкретных методов мониторинга и управления ценностью, и значительная часть выгод реализуется на этапе постпроекта, когда невозможно повлиять на продукт.

Таким образом, существует потребность в подходе к управлению ценностью в организационных ИТ-проектах, в котором бы использовались инструменты управления, доступные во время планирования проекта и стадий жизненного цикла, на которых происходит поставка продукта. Такие инструменты можно клас-

сифицировать на инструменты, ориентированные на повышение точности планирования содержания и анализа требований к продукту, и на инструменты, ориентированные на поставку запланированной ценности в условиях изменений в содержании. Первая группа инструментов имеет ограниченную эффективность, так как заказчику невозможно точно установить соответствие образа продукта той ценности, которую он должен создать. В силу того, что точно спланировать содержание проекта не удастся, в течение проекта вносятся изменения, что создает риск расползания содержания, основной причиной которого называется недостаточно проработанный дизайн программного обеспечения на ранних стадиях проекта [16]. Однако увеличение затрат на планирование в ИТ-проектах не приводит к снижению этого риска [11]. Настоящая статья описывает подход, основанный на инструментах второй группы, т.е. ориентированных на использование изменений в содержании для управления поставкой ценности в проекте.

Проблемы контроля ценности в организационных ИТ-проектах

Если рассматривать ценность проекта как приращение ценности (стоимости) бизнеса, получаемое вследствие использования в операционной деятельности результатов проекта, то эта ценность создается на стадии постпроекта. Таким образом, у руководителя проекта и проектного совета нет управленческих механизмов для управления непосредственным созданием бизнес-ценности. В организационных ИТ-проектах процесс создания ценности в общем виде состоит из следующих этапов (рис. 1): поставка предварительных результатов (разработанное ПО,



Рис. 1. Распределение факторов, влияющих на бизнес-ценность, по жизненному циклу ИТ-проекта

инструкции, обучающие программы, регламенты бизнес-процессов и т.п.), поставка промежуточных результатов (автоматизация отдельных бизнес-процессов), интеграция результатов или поставка организационно-технического решения и выполнение мероприятий постпроекта. На рисунке процесс создания ценности показан в привязке к этапам жизненного цикла проекта.

Пересекающиеся этапы разработки технического решения и выполнения организационного изменения означают, что результаты разработки по мере из поставки внедряются в операционную деятельность и изменяют процессы компании. В традиционном жизненном цикле проекта организационных изменений на базе внедрения информационных систем эти этапы выполняются последовательно. Процессы на стороне заказчика включают определение видения и разработку конфигурации решения, после чего на стороне подрядчика выполняется разработка технического решения, соответствующего конфигурации, затем техническое решение внедряется, после чего выполняется собственно организационное изменение [23]. Чтобы иметь возможность управлять изменениями содержания по результатам мониторинга ценности, необходимо передавать результаты разработки в операционную деятельность по частям, что позволит выявлять отклонения от запланированной ценности до завершения проекта и выполнять корректирующие действия. Для этого в настоящей работе предлагается использовать сервисную модель продукта проекта, представленную на рис. 2.

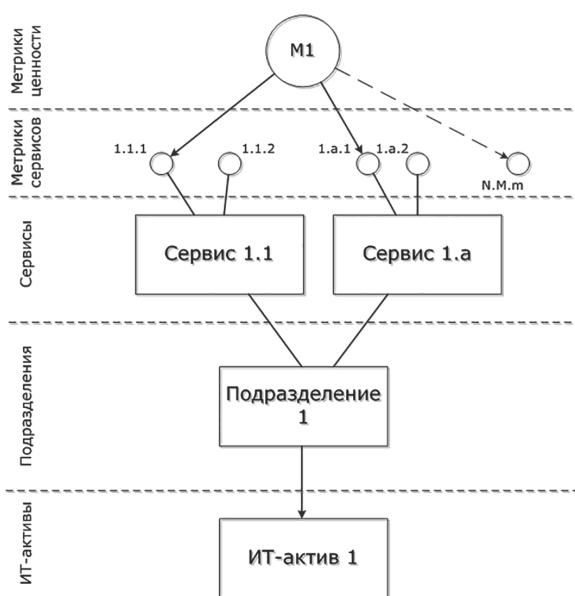


Рис. 2. Сервисная модель продукта проекта

Продукт проекта рассматривается не как внедренная система автоматизации (поставка ее по частям, как правило, невозможна), а как набор сервисов, предоставляемых подразделением, использующим в своей деятельности создаваемые ИТ-активы (ИТ-службой, офисом управления проектами, системой менеджмента качества и т.п.), организации. Данная модель основывается на модели сервисов *ITSM* [1; 15]. В описываемом подходе деятельность (или часть деятельности) любого подразделения, использующего в своей деятельности ИТ-актив, может быть представлена набором сервисов. Собственно, в *ITSM* описан только подход к организации работы ИТ-службы организации. Применения сервисной модели к работе иных подразделений организаций достаточно редки. Заслуживает внимание, например, идея организации по сервисной модели работы проектного офиса [3].

Другой существенной проблемой ИТ-проектов является проблема определения метрик ценности для мониторинга. Метрики должны быть:

- значимыми для бизнеса или программы, если проект выполняется в рамках программы;
- измеримыми на организационном уровне;
- достаточно чувствительными для того, чтобы отразить изменения, ожидаемые после выполнения проекта.

Кроме того, метрики должны существовать вне зависимости от наличия / отсутствия ИТ-проекта. Если параметр процесса или услуги может быть измерен только при применении продукта проекта, то этот параметр не может служить метрикой ценности. Это требование выдвигается в силу того, что метрики должны использоваться для оценки успешности проекта путем сравнения состояния до и после выполнения проекта. При использовании сервисной модели метрики качества отдельных разработанных или измененных в проекте сервисов составляют вместе метрики ценности проекта, на основании мониторинга которых с использованием метода реализованной ценности [2] и принимаются решения об изменении содержания. Для применения метода реализованной ценности используемые для мониторинга метрики должны быть возрастающими, т.е. большее значение метрики должно соответствовать более предпочтительному для организации состоянию измеряемого фактора ценности.

Метод реализованной ценности основывается на методе освоенного объема и в дополнение к индексам выполнения стоимости и сроков вводит

дополнительный индекс реализации ценности PPI . Компоненты данного вектора — коэффициенты эффективности, рассчитанные для каждой отдельной метрики ценности. Для расчета коэффициентов используются переменные: плановые значения метрик ценности в зависимости от освоенного объема — $PP(EV)$ и фактические значения метрик в данный момент времени — AP .

Зависимость от освоенного объема означает, что сопоставление плановых и фактических значений будет осуществляться не на данный момент времени вне зависимости от объема выполненных работ, а по значениям соответствующим освоенному объему. Таким образом, метод позволяет рассчитать эффективность выполненных задач проекта с точки зрения их вклада в реализацию ценности проекта.

$$PPI_i = \frac{AP_i}{PP_i(EV)}, \quad (1)$$

где i — индекс метрики ценности;

AP_i — реализованная ценность, оцениваемая i -й метрикой ценности, в единицах измерения соответствующей метрики;

PP_i — плановая ценность, оцениваемая i -й метрикой ценности, в единицах метрики;

EV — освоенный объем, в рублях; или в привязке к текущему времени

$$PPI_i(t) = \frac{AP_i(t)}{PP_i(t - \delta)}, \quad (2)$$

где δ — текущее отставание от выполнения по плану.

Эффективность управления ценностью оценивается по значениям коэффициентов реализации ценности (компонент вектора PPI) и их сопоставления с коэффициентом выполнения сроков (рис. 3). Оценка осуществляется в контрольных точках на границах этапов или других точках, определяемых планом управления содержанием, в которых разрешено внесение изменений. Ниже приводится трактовка сочетаний значений PPI_i и SPI .

$SPI = 1, PPI_i \geq 1$. Нормальная ситуация. Коррекция не требуется.

$SPI < 1, PPI_i \geq 1$. Отставание от плана в выполнении работ проекта, требуются корректирующие действия для повышения производительности. При этом реализация ценности происходит в соответствии с планом. При этом может наблюдаться и отставание от плана реализации ценности в привязке

ко времени. Метод реализованной ценности, в отличие от метода освоенного объема, в такой ситуации позволяет сделать вывод о том, что вмешательство в процессы реализации ценности не требуется.

$SPI > 1, PPI_i < 1$. Опережение графика выполнения проекта, но отставание от плана реализации ценности по одному из каналов. Такое сочетание коэффициентов эффективности является индикатором того, что в действительности проект отстает от графика, и из-за проблем в проекте выполняются не те работы, которые должны выполняться в текущих итерациях поставки продукта, а работы из будущих итераций, которые в данный момент не могут создавать ценность.

$SPI > 1, PPI_i \geq 1$. Выполняются работы следующего этапа. Коэффициент реализации ценности в таком случае может давать искаженную информацию. Если текущий этап завершен, следует его закрыть и внести изменения в базовый план проекта, в противном случае сфокусироваться на выполнении работ текущего этапа.

$SPI < 1, PPI_i < 1$. Имеет место отставание от плана как по временным параметрам, так и по реализации ценности по соответствующему каналу. Метод реализованной ценности позволяет установить факт отставания в реализации ценности и запланировать изменения в содержании проекта на более ранних этапах выполнения проекта, когда их стоимость ниже.

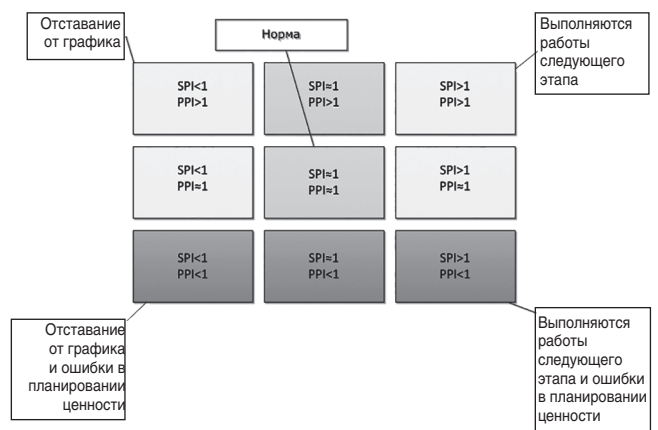


Рис. 3. Анализ значений коэффициентов реализации ценности PPI и выполнения сроков SPI

Можно сформулировать следующие методические рекомендации по управлению ценностью в организационном ИТ-проекте:

- определить метрики ценности так, чтобы они описывали значимые для бизнеса изменения параметров основных процессов;

- определить продукт проекта как набор сервисов, предоставляемых с помощью внедряемых ИТ-активов;
- разделить поставку продукта проекта на этапы, каждый из которых включает внедрение или изменение одного или нескольких сервисов;
- планировать изменение метрик ценности в течение жизненного цикла проекта в соответствии с планом внедрения сервисов;
- отслеживать изменение метрик ценности в сравнении с планом и вносить изменения в план поставки сервисов по данным мониторинга.

При внесении изменений в содержание проекта по данным обратной связи о реализации бизнес-ценности следует контролировать риск расползания содержания. Далее в статье излагается метод анализа устойчивости системы управления проектом, который может использоваться для оценки указанного риска.

Устойчивость системы управления проектом при управлении изменениями в содержании

Под устойчивостью будем понимать способность системы управления проектом обеспечивать создание запланированной бизнес-ценности в условиях установленных ограничений по срокам и затратам при наличии изменений в содержании проекта. При этом устойчивость является внутренним свойством системы управления. Изменения же, по отношению к которым оценивается устойчивость, могут быть как внутренним, так и внешним фактором. Внешним фактором изменения будут в том случае, если проекты разработки ИТ-решения и выполнения организационного изменения на его основе управляются отдельно [5]. В случае же, если команды на стороне заказчика и подрядчика совместно работают над созданием бизнес-ценности, и организационная и техническая части проекта охвачены общей обратной связью [18], то изменения в содержании проекта могут считаться внутренним фактором, и их интенсивность обусловлена качеством процессов мониторинга и управления. Таким образом, для излагаемого здесь подхода с применением метода реализованной ценности будем считать изменения содержания внутренним фактором.

Для анализа риска расползания требований необходимо вычислить величину запаса устойчивости: превышение доступного управленческого резерва над ожидаемыми изменениями содержания. На

рисунке показаны факторы, влияющие на запас устойчивости (рис. 4).

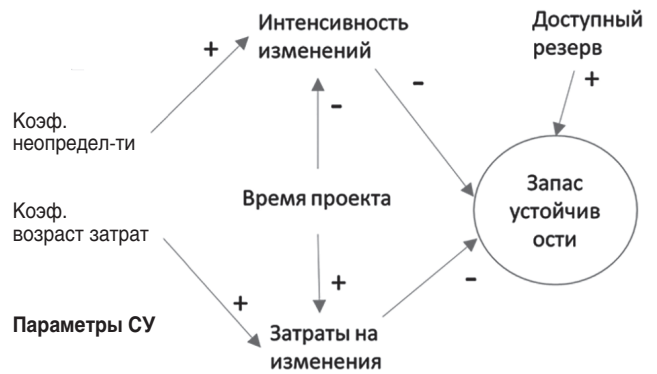


Рис. 4. Факторы, влияющие на запас устойчивости

Видно, что величина запаса устойчивости непостоянна во времени. Ее следует рассчитывать по формуле:

$$R(t) = \rho(t) - Q(t) = \rho(t) - \tilde{F}(t) \cdot C(t), \quad (3)$$

где $R(t)$ — запас устойчивости, руб.;

$\rho(t)$ — доступный в данный момент времени резерв на изменения в содержании, выраженный в денежных единицах, руб.;

$Q(t)$ — ожидаемые затраты на изменения, руб.;

$\tilde{F}(t)$ — ожидаемые изменения в рассматриваемый момент времени, выраженные в тех единицах, в которых оценивается объем проекта (ед.);

$C(t)$ — затраты на выполнение изменения объемом в одну единицу в рассматриваемый момент времени, руб./ед.

Для показателя запаса устойчивости системы управления сформулированы следующие свойства и граничные значения, используемые для оценки риска расползания содержания:

- 1) запас устойчивости изменяется в процессе выполнения проекта, так как изменяется как вероятность внесения изменений, так и требуемые трудозатраты на их реализацию;
- 2) положительное значение запаса устойчивости означает устойчивость системы управления к ожидаемым отклонениям, отрицательное — неустойчивость, при равенстве нулю система управления находится на границе устойчивости;
- 3) значительное снижение рассчитанного запаса устойчивости за короткое время означает то, что в процессах управления изменениями произошли изменения. Пороговые значения для скорости

снижения показателя устойчивости определяют политиками и стандартами компании в области управления рисками и уточняются при накоплении опыта реализации проектов.

Запас устойчивости системы управления зависит от интенсивности изменений и от трудозатрат на их выполнение. Эти факторы являются внутренними свойствами системы управления проектом и могут быть определены по ретроспективным данным о ранее выполненных изменениях в содержании. Интенсивность изменений зависит от принятого метода управления ценностью, качества процессов планирования и мониторинга, затраты на изменения зависят от их сложности и связности архитектуры как технического, так и организационного решения. Ожидаемый объем изменений определяется закономерностью, известной как «конус Боэма», т.е. снижения неопределенности и связанного с ней объема вносимых изменений по мере выполнения работ проекта [6]. Затраты на выполнение изменений одного и того же объема, напротив, возрастают на более поздних стадиях проекта, так как повышается число связей целевого архитектурного модуля с другими модулями, и это порождает изменения в связанных модулях.

При сравнении стоимости изменений, вносимых в разные периоды времени, возникает проблема выбора единиц измерения для оценки их объема. Денежные единицы, традиционно применяющиеся для оценки объема проекта, не подходят, так как стоимость выполнения изменения зависит от времени его внесения, а не только от его сложности. В ИТ-отрасли также имеют распространение относительные оценки объема, функциональные единицы. Их основным свойством является то, что два изменения, оцененные в равное количество функциональных единиц и внесенные в один момент времени, будут выполнены с равными затратами.

Получена следующая формула для расчета трудозатрат на изменения в зависимости от времени проекта (обоснование ее приводится ниже):

$$C(t) = C_0 \cdot r^{(2 \cdot \tau)} \quad (4)$$

где $C(t)$ — полные трудозатраты на изменение объемом в одну функциональную единицу, руб./ф.е.;
 r — коэффициент возрастания трудозатрат в течение жизненного цикла проекта, безразмерная величина;

C_0 — выраженные в денежных единицах трудозатраты на выполнение изменения такого же объ-

ема, если изменение внесено в начале проекта, до начала разработки продукта, руб.

Уточненная с учетом рассматриваемых закономерностей формула запаса устойчивости:

$$R(t) = \rho(t) - \varepsilon \cdot (1 - \tau) \cdot S_0 \cdot C_0 \cdot r^{(2 \cdot \tau)} \quad (5)$$

где $\rho(t)$ — доступный в момент времени резерв на изменения, руб.;

ε — коэффициент интенсивности изменений, обусловленных неопределенностью требований, безразмерная величина;

S_0 — объем проекта в соответствии с базовым планом, выраженный в функциональных единицах;

C_0 — относительные затраты на изменения, если они вносятся на старте проекта, руб./ф.е.;

r — коэффициент возрастания трудозатрат на изменения, вносимые на поздних стадиях проекта.

Коэффициент интенсивности изменений, обусловленных неопределенностью требований в контрольной точке, рассчитывается по ретроспективным данным по формуле:

$$\varepsilon(t_i) = \frac{S_i}{S_0} \cdot \frac{T}{t_i - t_{i-1}}, \quad (6)$$

где t_i — время наступления i -й контрольной точки;

S_i — общий объем изменений в функциональных единицах, утвержденных в интервал времени $(t_{i-1}, t_i]$;

S_0 — объем требований к продукту в функциональных единицах в соответствии с базовым планом.

Коэффициент возрастания трудозатрат на поздних стадиях проекта:

$$r_i = \sqrt[2 \cdot \tau]{\frac{C_i}{C_0}} = \sqrt[2 \cdot \tau]{\frac{C_i}{P}}, \quad (7)$$

где C_i — затраты на одну функциональную единицу по результатам выполнения i -го изменения, руб./ф.е.;

C_0 — затраты на одну функциональную единицу по результатам выполнения изменения, внесенного на старте проекта, руб./ф.е., равны продуктивности команды P (руб./ф.е.) при реализации функционала из базового плана;

τ_i — относительное время проекта, в которое внесено i -е изменение.

$$C_0 = P = \frac{V_0}{F_0}, \quad (8)$$

где V_0 — стоимость проекта в соответствии с базовым планом, руб.;

F_0 — объем функционала продукта в соответствии с базовым планом, ф.е.

Обоснование характера возрастания трудозатрат на изменения в течение жизненного цикла проекта

Зависимость трудозатрат на изменения от жизненного цикла обусловлена тем, что по мере разработки программного решения растет число связей между модулями технического решения и возрастает вероятность того, что вносимое в модуль изменение затронет не только целевой модуль, но и несколько смежных. Эта вероятность тем больше, чем выше связность архитектуры программного обеспечения.

Связность программного обеспечения определяется коэффициентом CF (*Coupling Factor*), который определяется используемой технологией программного обеспечения и не зависит от размера программного решения [8].

$$CF = \frac{AC}{MPC}, \tag{9}$$

где AC (*actual coupling*) — число связей между модулями программного обеспечения (суммарно входящих и исходящих);

MPC (*maximal possible coupling*) — максимально возможное число связей между модулями (все модули связаны со всеми остальными как входящими, так и исходящими связями).

Введем в рассмотрение аналогичный коэффициент, определяющий связность одного модуля программного обеспечения:

$$cf = \frac{ac}{mpc} = \frac{ac}{2 \cdot (m-1)}, \tag{10}$$

где cf — связность одного модуля программного обеспечения;

ac — фактическое число связей модуля программного обеспечения, как входящих, так и исходящих;

mpc — максимально возможное число связей модуля (модуль связан со всеми модулями архитектуры программного обеспечения, кроме самого себя, как входящими, так и исходящими связями).

Таким образом, ожидаемое число связей одного модуля программного обеспечения будет определяться по формуле:

$$ac = 2 \cdot cf \cdot (m - 1) \tag{11}$$

Суммарные трудозатраты на внесение изменений будут определяться трудозатратами на изменение целевого модуля и в случае необходимости —

в смежные модули. Таким образом, для оценки суммарных затрат необходимо оценить вероятность того, что возникнет потребность во внесении изменений в модули, смежные целевому.

Рассмотрим модель архитектуры программного обеспечения (рис. 5). Допустим, целевое изменение вносится в модуль A . Данное изменение может повлечь за собой:

- вторичные изменения первого порядка в модуле B по связи $b1$, в модуле C по связи $c1$ и в модуле D по связи $d1$;
- вторичные изменения второго порядка в модуле C по связи $c2$ и в модуле D по связи $d2$. Эти изменения являются также вторичными изменениями первого порядка по отношению к изменениям в модуле B ;
- вторичные изменения третьего порядка в модуле D по связи $d3$. Эти изменения являются также вторичными изменениями первого порядка по отношению к изменениям в модуле C и изменениями второго порядка по отношению к изменениям в модуле B .

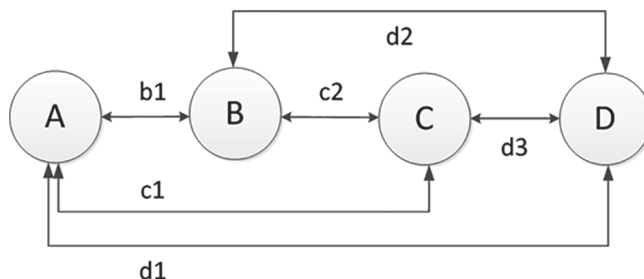


Рис. 5. Модель архитектуры технического решения

Распределение связей в пределах архитектуры равномерное, т.е. вероятность того, что внесение изменение в любой модуль повлечет вторичные изменения по каждой архитектурной связи, может рассматриваться одинаковой для всех изменений (p^s).

Таким образом, ожидаемые затраты на изменения в модуле A с учетом всех вторичных изменений следует рассчитывать по формуле:

$$C_{total} = e_0 + e_1 + e_2 + e_3 + \dots, \tag{12}$$

где C_{total} — суммарные затраты на изменения;
 $e_0 = \sigma \cdot p_0$ — ожидаемый объем изменений в целевом модуле;

σ — коэффициент затрат на реализацию функционала (руб.), равный затратам на изменение объемом в одну функциональную единицу в целевой модуль при отсутствии вторичных изменений;

$p_0 = 1$ — вероятность внесения изменений в целевой модуль;

e_i ($i = 1, 2, 3...$) — вероятность вторичных изменений i -го уровня.

Вероятность вторичных изменений первого уровня по одной связи равна:

$$p_1 = p_0 \cdot p^s. \quad (13)$$

Ожидаемые вторичные изменения по всем связям первого уровня:

$$e_1 = \sigma \cdot p_1 \cdot ac = \sigma \cdot 2 \cdot cf \cdot (m-1) \cdot p^s. \quad (14)$$

Вероятность вторичных изменений второго уровня по одной связи с учетом того, что они возможны только при условии наличия изменений первого уровня, будет определяться по формуле:

$$p_2 = \frac{p_1 \cdot p^s}{2}. \quad (15)$$

Вероятность делится на 2 в связи с тем, что модули, имеющие связи второго уровня с целевым, могут также иметь связи первого уровня. Поскольку распределение связей в архитектуре равномерное и изменение в один и тот же модуль, как правило, не вносится одновременно как изменение первого порядка и изменение второго порядка, то вероятность должна делиться на число исходов.

Тогда ожидаемые вторичные изменения второго уровня:

$$e_2 = S \cdot p_2 \cdot ac^2 = \frac{\sigma \cdot (2 \cdot cf \cdot (m-1) \cdot p^s)^2}{2}. \quad (16)$$

Объем ожидаемых вторичных изменений третьего порядка в соответствии с той же логикой определяется по формуле:

$$e_3 = \frac{S \cdot p_2 \cdot ac^3 \cdot p^s}{3} = \frac{\sigma \cdot (2 \cdot cf \cdot (m-1) \cdot p^s)^3}{2 \cdot 3}. \quad (17)$$

Тогда формула для полных затрат на изменение примет вид:

$$C_{total} = \sigma \cdot \left(1 + 2 \cdot cf \cdot (m-1) \cdot p^s + \frac{(2 \cdot cf \cdot (m-1) \cdot p^s)^2}{2} + \frac{(2 \cdot cf \cdot (m-1) \cdot p^s)^3}{2 \cdot 3} + \dots \right) = S \cdot \left(1 + \sum_{i=1}^N \frac{(2 \cdot cf \cdot (m-1) \cdot p^s)^i}{i!} \right), \quad (18)$$

где N — максимальная глубина изменений, вызываемых изменениями в целевом модуле.

Данная формула сворачивается по формуле ряда Тейлора в экспоненциальную функцию:

$$C_{total} = \sigma \cdot e^{2 \cdot cf \cdot (m-1) \cdot p^s}. \quad (19)$$

Рост числа модулей технического решения в течение проекта для проекта разработки программного обеспечения можно рассматривать как равномерный.

$$m(t) = \tau \cdot M, \quad (20)$$

где $\tau = \frac{t}{T}$ — относительное время проекта, $\tau = 0$ в начале проекта, $\tau = 1$ в момент его закрытия;

t — время проекта в днях;

T — длительность проекта в днях;

M — число модулей в техническом решении в момент закрытия проекта.

$$C_{total}(t) = \sigma \cdot e^{2 \cdot cf \cdot (\tau \cdot M - 1) \cdot p^s} = \sigma \cdot \frac{e^{2 \cdot cf \cdot (\tau \cdot M) \cdot p^s}}{e^{2 \cdot cf \cdot p^s}}. \quad (21)$$

Введем дополнительные обозначения для констант и переменных, определяемых технологией и масштабом проекта:

$r = e^{cf \cdot p^s \cdot M}$ — коэффициент возрастания трудозатрат в течение жизненного цикла проекта;

$C_0 = \frac{\sigma}{e^{2 \cdot cf \cdot p^s}}$ — трудозатраты на выполнение изменения такого же объема, если изменение внесено в начале проекта, до начала разработки продукта.

Тогда

$$C_{total} = C_0 \cdot r^{2 \cdot \tau}. \quad (22)$$

Верификация полученной формулы для расчета трудозатрат на изменения. Для верификации используются данные о трудозатратах на исправление дефектов, внесенных на этапе разработки требований и обнаруженных на разных этапах жизненного цикла проекта [4], полученные на основе анализа данных из различных статей в специализированных журналах и материалах конференций [7; 10; 12–14; 17; 25; 27]. В качестве эталонных данных для сравнения используются данные о затратах на исправление дефектов, так как процесс исправления дефектов в ИТ-проектах полностью идентичен процессу внесения изменений в требования. Кроме того, есть методы управления проектами, которые не различают эти процессы. Так, например, в *PRINCE2* внесение изменений в содержание и исправления отклонений от спецификации требований осуществляется в рамках одного процесса управления инцидентами [20].

Данные, рассчитанные по полученной выше формуле затрат на изменения, приведены в табли-

це сравнения (табл. 1) и сопоставляются с эталонными данными из анализа Макконнелла (нижняя строка таблицы). В таблице приводятся относительные значения трудозатрат. Трудозатраты на изменения, вносимые до выполнения проекта, приняты за 1. Затраты рассчитаны для коэффициентов возрастания трудозатрат от 2,5 до 4. В табл. 1 выделены результаты, совпадающие с эмпирическими данными, представленными Макконнеллом.

Таблица 1

Сравнение расчетных и эталонных значений трудозатрат на изменения

Коэффициент r	Время обнаружения дефекта				
	Выработка требований	Проектирование архитектуры $\tau = 0,25 \dots 0,5$	Конструирование $\tau = 0,5 \dots 0,9$	Тестирование системы $\tau = 0,9 \dots 1$	После выпуска продукта $\tau \approx 1$
$r = 2,5$	1	1,58–2,5	2,5–5,2	5,2–6,25	6,25
$r = 3$	1	1,73–3	3–7,22	7,22–9	9
$r = 3,5$	1	1,87–3,5	3,5–9,54	9,54–12,25	12,25
$r = 4$	1	2–4	4–12,13	12,13–16	16
Эталонные значения	1	3	5–10	10	10–100

Экспоненциальной аппроксимацией приведенных выше эмпирических данных по методу наименьших квадратов в пакете *Microsoft Excel* была получена следующая функция (рис. 6).

$$C_e = e^{2,604 \cdot \tau} = 3,6772 \cdot \tau. \quad (23)$$

Достоверность аппроксимации, выраженная коэффициентом детерминации $R^2 = 0,9621$.

Точки для аппроксимации были выбраны следующим образом:

- точка выработки требований $\tau = 0$, $C_e = 1$ соответствует началу проекта. Именно в момент начала проекта любые изменения в содержание стоят столько же, сколько стоят те же работы в первоначальном содержании;
- точка выработки архитектуры $\tau = 0,375$, $C_e = 3$ поставлена на середину интервала (короткий интервал, в эмпирических данных для него приведено одно число);
- точки конструирования $\tau = 0,5$, $C_e = 5$ и $\tau = 0,9$, $C_e = 10$ поставлены на границах интервала (длинный интервал, в эмпирических данных для него приведен диапазон);
- точка тестирования $\tau = 0,95$, $C_e = 10$ поставлена на середину интервала (короткий интервал).

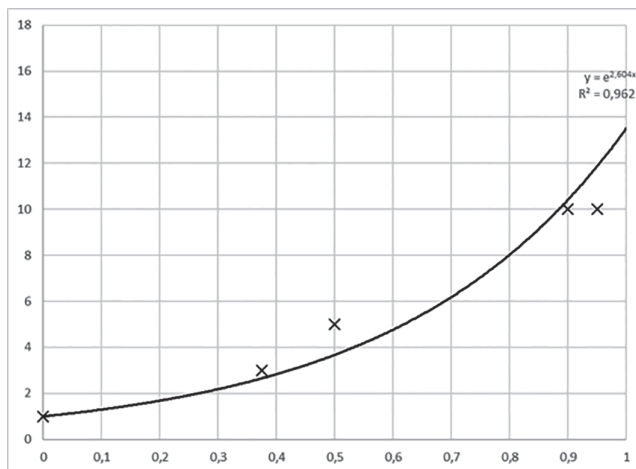


Рис. 6. Аппроксимация эмпирических данных о возрастании трудозатрат экспоненциальной функцией

Можно сделать вывод о том, что результаты расчета затрат на изменения, выполненные по выведенной формуле, соответствуют эмпирическим данным, являющимся обобщением результатов многих исследований, проведенных в отрасли разработки программного обеспечения. Характер возрастания трудозатрат, описываемый полученной формулой при коэффициенте $r = 3,7$, ближе всего соответствует эмпирическим данным. Коэффициент детерминации, равный 0,96, позволяет считать экспоненциальную аппроксимацию допустимой.

Следует отметить, что несмотря на сохранение экспоненциального характера возрастания трудозатрат на изменения в течение жизненного цикла проекта при применении различных технологий разработки программного обеспечения, скорость возрастания зависит от связности архитектуры и масштаба технического решения (в таблице в качестве эталонных приведены средние данные по отрасли).

Отдельно следует также пояснить верхнюю границу диапазона трудозатрат на изменения после выпуска продукта (как видно из таблицы, они могут возрасти до ста раз по сравнению со стоимостью изменений, вносимых на этапе разработки требований). Это объясняется тем, что после выпуска продукта число связей продукта продолжает возрастать, но это внешние по отношению к продукту связи, обусловленные воздействием продукта на бизнес-процессы, регламенты организации, инструкции, деятельность сотрудников и т.п.

Заключение

В статье представлен подход к построению системы управления созданием ценности в организа-

ционных ИТ-проектах, основанный на применении сервисного подхода к определению продукта проекта, поэтапном внедрении результатов проекта, мониторинге ценности и управлении изменениями в содержании проекта на основе мониторинга. Данный подход позволяет снизить затраты на выполнение проекта за счет внесения необходимых изменений в содержание на более ранних этапах, когда их выполнение требует меньших затрат, снизить риск непоставки запланированной ценности за счет поэтапного внедрения результатов проекта в операционную деятельность и контроля ценности, а также риск расползания требований за счет мониторинга запаса устойчивости.

Применение сервисного подхода позволяет связать метрики бизнес-ценности с метриками качества продукта, а также организовать поставку продукта проекта по частям, что необходимо для реализации обратной связи в проекте на основе изменений содержания.

Метод реализованной ценности дополнен по сравнению с ранее опубликованной версией кри-

териями эффективности реализации ценности. Приведена интерпретация значений индекса реализации ценности в сопоставлении со значениями индекса выполнения сроков метода освоенного объема. Применение метода позволяет снизить затраты на изменения и повысить их обоснованность с точки зрения создаваемой бизнес-ценности благодаря раннему выявлению отклонений от плана поставки ценности, в том числе в условиях отставания от графика, а в условиях опережения графика выявить факт выполнения работ поздних этапов, не создающих на ценности на текущем этапе.

В работе дано обоснование формулы оценки запаса устойчивости системы управления проектом, при помощи которой может быть выявлен риск расползания требований, что актуально при использовании изменений содержания проекта в качестве инструмента управления поставкой бизнес-ценности. С той же целью выведена и верифицирована формула, подтверждающая экспоненциальный характер возрастания затрат на изменения в зависимости от времени их внесения.

Литература

1. Аксенов Е. Аутсорсинг: 10 заповедей и 21 инструмент [Текст] / Е. Аксенов, И. Альтшулер. — СПб.: Питер, 2009. — 464 с.
2. Зуйков К.А. Обеспечение устойчивости системы управления ИТ-проектами [Текст] / К.А. Зуйков // Российский журнал управления проектами. — 2016. — № 1.
3. Козодаев М.А. Практика построения проектных офисов (Часть 2) [Текст] / М.А. Козодаев // Управление проектами и программами. — 2012. — № 4. — С. 270–282.
4. Макконнелл С. Совершенный код. Мастер-класс [Текст] / С. Макконнелл. — М.: Русская редакция, 2010. — 896 с.
5. Пащенко Д.С. О факторах, тормозящих развитие российских ИТ-компаний в фазе зрелости [Текст] / Д.С. Пащенко // Менеджмент и бизнес-администрирование. — 2013. — С. 123–128.
6. Boehm V. Software engineering economics // 1981.
7. Boehm V. Balancing agility and discipline: A guide for the perplexed // Software Engineering Research And Applications; Lecture Notes in Computer Science: Springer-Verlag Berlin, 2004.
8. Briand L.C., Daly J.W., Wust J.K. A unified framework for coupling measurement in object-oriented systems // IEEE Transactions on software Engineering. 1999. Т. 25. № 1. С. 91–121.
9. Croteau A.-M., Bergeron F. An information technology trilogy: business strategy, technological deployment and organizational performance // The journal of strategic information systems. 2001. Т. 10. № 2. Pp. 77–99.
10. Dunn R. Software defect removal. New York, NY, USA, 1984. McGraw-Hill, Inc.
11. Eisenhardt K.M., Tabrizi B.N. Accelerating adaptive processes: Product innovation in the global computer industry // Administrative science quarterly. 1995. Pp. 84–110.
12. Fagan M.E. Design and code inspections to reduce errors in program development // Pioneers and Their Contributions to Software Engineering: Springer, 2001. Pp. 301–334.
13. Grady R.B. An Economic Release Decision Model: Insights into Software Project Management. 1999.
14. Humphrey W.S., Snyder T.R., Willis R.R. Software process improvement at Hughes Aircraft // IEEE Software. 1991. Т. 8. № 4. Pp. 11–23.
15. Knapp D. ITSM Process Design Guide. Fort Lauderdale, FL: J. Ross Publishing Inc., 2010. 255 p.
16. Kuprenas J.A., Nasr E.B. Controlling design-phase scope creep // AACE International Transactions. 2003. P. CS11.
17. Leffingwell D. Calculating the Return on Investment from More Effective Requirements Management // American Programmer. 1997. Т. 10. № 4. Pp. 13–16.
18. Nesheim T., Hunskaar H.M. When employees and external consultants work together on projects: Challenges of knowledge sharing // International Journal of Project Management. 2015. Т. 33. № 7. Pp. 1417–1424.
19. Nikula U. и др. Empirical validation of the Classic Change Curve on a software technology change project // Information and Software Technology. 2010. Т. 52. № 6. Pp. 680–696.
20. OGC. Managing Successful Projects with PRINCE2: The Stationery Office, 2009. 342 p.
21. Ohara S. A Guidebook for Project and Program Management for Enterprise Innovation (P2M). 2005. Project Management Association of Japan. 93 p.
22. PMI. Руководство к своду знаний по управлению проектами (руководство PMBoK) // Project Management Institute. — 2013. — Вып. 5. — 586 с.

23. Portougal V., Sundaram D. Business processes: operational solutions for SAP implementation. Hershey, PA: IRM Press, 2006. 329 p.

24. Rivard S., Raymond L., Verreault D. Resource-based view and competitive strategy: An integrated model of the contribution of information technology to firm performance // The Journal of Strategic Information Systems. 2006. T. 15. № 1. Pp. 29–50.

25. Shull F. и др. What we have learned about fighting defects // Software Metrics, 2002. Proceedings. Eighth IEEE Symposium on. IEEE, 2002. Pp. 249–258.

26. Van Rensburg A.C.J. An object-oriented architecture for business transformation // Computers & industrial engineering. 1997. T. 33. № 1. Pp. 167–170.

27. Willis R.R. Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process // Software Engineering Institute. 1998. № 6.

References

1. Aksenov E., Al'tshuler I. *Autsorsing: 10 zapovedey i 21 instrument* [Outsourcing: 10 commandments and 21 tools]. St. Petersburg, Piter Publ., 2009. 464 p.
2. Zuykov K.A. Obespechenie ustoychivosti sistemy upravleniya IT-proektami [Ensuring the sustainability of the IT project management system]. *Rossiyskiy zhurnal upravleniya proektami* [Russian Journal of Project Management]. 2016, I. 1.
3. Kozodaev M.A. Praktika postroeniya proeknykh ofisov [The practice of building design offices]. *Upravlenie proektami i programmami* [Project and program management]. 2012, I. 04(32), pp. 270–282.
4. Makkonnell S. *Sovershenny kod. Master-klass* [Perfect code. Master Class]. Moscow, Russkaya redaktsiya Publ., 2010. 896 p.
5. Pashchenko D.S. O faktorakh, tormozyashchikh razvitie rossiyskikh IT-kompaniy v faze zrelosti [About the factors hindering the development of Russian IT companies in the maturity phase]. *Menedzhment i biznes-administrirovanie* [Management and business administration]. 2013, pp. 123–128.
6. Boehm B. Software engineering economics // 1981.
7. Boehm B. Balancing agility and discipline: A guide for the perplexed // *Software Engineering Research And Applications; Lecture Notes in Computer Science*: Springer-Verlag Berlin, 2004.
8. Briand L.C., Daly J.W., Wust J.K. A unified framework for coupling measurement in object-oriented systems // *IEEE Transactions on software Engineering*. 1999, V. 25, I. 1, pp. 91–121.
9. Croteau A.-M., Bergeron F. An information technology trilogy: business strategy, technological deployment and organizational performance // *The Journal Of Strategic Information Systems*. 2001, V. 10, I. 2, pp. 77–99.
10. Dunn R. *Software defect removal*. New York, NY, USA, 1984. McGraw-Hill, Inc.
11. Eisenhardt K.M., Tabrizi B.N. Accelerating adaptive processes: Product innovation in the global computer industry // *Administrative science quarterly*. 1995, pp. 84–110.
12. Fagan M.E. Design and code inspections to reduce errors in program development // *Pioneers and Their Contributions to Software Engineering*: Springer, 2001, pp. 301–334.
13. Grady R.B. *An Economic Release Decision Model: Insights into Software Project Management*, 1999.
14. Humphrey W.S., Snyder T.R., Willis R.R. Software process improvement at Hughes Aircraft // *IEEE Software*. 1991, V. 8, I. 4, pp. 11–23.
15. Knapp D. *ITSM Process Design Guide*. Fort Lauderdale, FL: J. Ross Publishing Inc., 2010. 255 p.
16. Kuprenas J.A., Nasr E.B. Controlling design-phase scope creep // *AACE International Transactions*. 2003. P. CS11.
17. Leffingwell D. Calculating the Return on Investment from More Effective Requirements Management // *American Programmer*. 1997, V. 10, I. 4, pp. 13–16.
18. Nesheim T., Hunskaar H.M. When employees and external consultants work together on projects: Challenges of knowledge sharing // *International Journal of Project Management*. 2015, V. 33, I. 7, pp. 1417–1424.
19. Nikula U. Empirical validation of the Classic Change Curve on a software technology change project // *Information and Software Technology*. 2010, V. 52, I. 6, pp. 680–696.
20. OGC. *Managing Successful Projects with PRINCE2: The Stationery Office*, 2009. 342 p.
21. Ohara S. *A Guidebook for Project and Program Management for Enterprise Innovation (P2M)*, 2005. Vyp. Project Management Association of Japan. 93 p.
22. *PMI. Rukovodstvo k svodu znaniy po upravleniyu proektami (rukovodstvo PMBoK)* [PMI. A Guide to the Recognition of Project Management (PMBOK Guide)]. Project Management Institute Publ., 2013, I. 5, 586 p.
23. Portougal V., Sundaram D. Business processes: operational solutions for SAP implementation. Hershey, PA: IRM Press, 2006. 329 p.
24. Rivard S., Raymond L., Verreault D. Resource-based view and competitive strategy: An integrated model of the contribution of information technology to firm performance // *The Journal of Strategic Information Systems*. 2006, V. 15, I. 1, pp. 29–50.
25. Shull F. и др. What we have learned about fighting defects // *Software Metrics, 2002. Proceedings. Eighth IEEE Symposium on*: IEEE, 2002, pp. 249–258.
26. Van Rensburg A.C.J. An object-oriented architecture for business transformation // *Computers & industrial engineering*. 1997, V. 33, I. 1, pp. 167–170.
27. Willis R.R. Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process // *Software Engineering Institute*. 1998, I. 6.