УДК 004.9
DOI: 10.15827/0236-235X.127.349-357

Дата подачи статьи: 30.01.19
2019. Т. 32. № 3. С. 349–357

О формализации функциональных требований в проектах по созданию информационных систем

*Р.Д. Гутгарц*¹, д.э.н., профессор, gutgarc@gmail.com

*Е.И. Провилков*¹, аспирант, provilkoff@gmail.com

¹ *Иркутский национальный исследовательский технический университет (ИРНИТУ),
г. Иркутск, 664074, Россия*

Проблемам управления ИТ-проектами посвящено достаточно много научных и прикладных исследований. Главное внимание в них уделяется финансовым аспектам и продолжительности проекта, а функциональные аспекты часто остаются за кадром. Объяснить это можно тем, что такие показатели, как деньги и время, с учетом различных рисков можно рассчитать при использовании соответствующих методик и алгоритмов. Между тем адекватного численного эквивалента для определения функциональности проекта до сих пор не существует.

При рассмотрении такого специализированного ИТ-проекта, как проектирование и разработка информационной системы или ее отдельного модуля, реализуемая в его рамках функциональность является основополагающим фактором, от которого зависят все другие показатели проекта. Однако функциональности в семантическом понимании этого слова по разным причинам в фундаментальной литературе и периодических научных изданиях уделяется недостаточно внимания. Основные интересы ученых и специалистов сосредоточены на управлении требованиями к информационной системе, включая функциональные. Формализация функциональных требований является предметом для обсуждения в научном сообществе и рассматривается в разных аспектах, но каких-либо унифицированных решений не предлагается.

В статье рассматриваются вопросы, связанные исключительно с функциональными требованиями, особенностями их изначального формулирования, представлением для обсуждения с заказчиком и с формализацией для программного воплощения в проекте.

Приводится краткий анализ подходов к формализации требований. Предложен авторский подход к формализации функциональных требований, который может быть распространен на определенные типы задач, включенные в ПО, представляющее информационную систему в виде программного продукта. Это может быть первым шагом в создании предпосылок для разработки алгоритмической составляющей, то есть для более корректного расчета трудоемкости проекта и, как следствие, более точного планирования его финансовых и временных затрат.

Ключевые слова: *ИТ-проект, проектирование информационных систем, функциональные требования к информационным системам, проблемы формулирования функциональных требований, формализация функциональных требований, унификация функциональных требований.*

Тематика требований к *информационным системам* (ИС) весьма разнообразна: функциональные требования, требования к техническому обеспечению, инструментальным средствам программирования и ведения БД, к безопасности и др. Кроме того, в данную тематику входит рассмотрение вопросов, которые могут обобщать такое понятие, как «управление требованиями», то есть их взаимозависимость, согласование, утверждение, оценку, редактирование и др. И каждый аспект таит в себе множество проблем, требующих отдельного анализа.

С течением времени острота такого рода проблем практически не уменьшается, а если и становится меньше, то совсем незначительно.

По-видимому, это обусловлено двумя основными причинами. Первая заключается в существенной терминологической разнице понятий, которыми оперируют представители заказчика (предметная область, то есть специализация деятельности) и разработчика функционального ПО (их предметная область всегда одна и та же – *информационные технологии* (ИТ)). По этой причине в команде исполнителя часто присутствуют аналитики и эксперты, хорошо понимающие специфику предметной области заказчика. Однако не каждая организация может позволить себе такое участие из-за ограниченный финансового характера. Практика показывает, что даже профессиональные бизнес-аналитики из команды разработчика не сразу

находят общий язык с представителями заказчика. Вторая причина заключается в неумении разработчика выявить у заказчика именно ту информацию, которая будет корректно отражать его функциональные требования к будущей системе, и в неумении заказчика сформулировать свои пожелания того, что в действительности должно быть реализовано в проектируемой системе.

Проблемы при взаимодействии заказчика и разработчика существуют с начала активного создания прикладных автоматизированных систем, о чем свидетельствует карикатура, опубликованная в информационном бюллетене вычислительного центра Лондонского университета в 1973 г. И эти проблемы до сих пор не утратили своей актуальности. В настоящее время имеется современная интерпретация упомянутой иллюстрации, известная под названием *Бизнес-проект «Качели»* [1]. Хотя по характеру рисунок является шуточным, он отражает принципиально важный аспект в процессе проектирования и разработки ИС: результат выполнения проекта может оказаться невостребованным, если в нем не реализована нужная заказчику функциональность.

Кроме того, говоря об ИС, часто понимают под ней микроИС, например, ИС заказа товаров. Бесспорно, ИС в таком виде имеет право на существование. Но, согласно классическому представлению системы, она всегда состоит из отдельных подсистем (модулей). Модульная структура является, например, одной из основополагающих характеристик ERP-систем (систем планирования ресурсов предприятия), очень распространенных и востребованных в настоящее время. Поэтому процесс рассмотрения требований к ИС, предназначенной для заказа товаров, будет, очевидно, отличаться от более крупной ИС, ориентированной, например, на управление каким-либо предприятием (организацией) или его отдельным подразделением.

Как известно, именно функциональные требования являются первичными в проекте по созданию ИС. Все прочие типы требований (включая их количество) напрямую зависят от функциональных. Поэтому очень важно, во-первых, учесть как можно больше такого рода требований на самой ранней стадии проекта, во-вторых, корректно сформулировать эти требования и впоследствии утвердить.

Статистические данные за 2008 год, опубликованные компанией Standish Group, свидетельствуют о том, что из 30 тысяч проанализи-

рованных программных проектов только 1/3 (32 %) завершились успешно, почти половина проектов (44 %) завершились с проблемами (главным образом, превысили бюджет и сроки) и почти четверть (24 %) полностью провалились [2].

В работе [3] отмечается, что при выявлении ошибок в требованиях на стадии начального формулирования стоимость их устранения будет в 5–10 раз меньше по сравнению с тем, когда эти требования трансформируются в программный код. Если же по каким-либо причинам ошибки на этих двух стадиях не были обнаружены и проявились только при сопровождении готовой к эксплуатации системы, стоимость их устранения превысит стоимость программирования уже в 20 раз. Авторы особо выделяют две категории ошибок, которые обнаруживаются на стадии проектирования:

- ошибки, возникающие на стадии технического проектирования при правильном формулировании требований;
- ошибки, возникающие на стадии технического проектирования по причине неудовлетворительного выявления требований в процессе предпроектного обследования.

Вторая категория ошибок, конечно, обходится дороже, поскольку требует возврата на предыдущие этапы проектирования. Причем эта процедура, как правило, является итерационной.

Фактически формулирование функциональных требований – второй этап в работе с требованиями. Первым этапом будет их выявление. Зависимость этапов при формулировании функциональных требований к ИС показана на рисунке 1.

Приведем несколько мнений о важности требований.

В [3] авторы выделяют следующие основные факторы, обуславливающие проблемы в будущих проектах по созданию ИС:

- недостаток исходной информации от клиента (13 % проектов);
- неполные требования и спецификации (12 % проектов);
- изменение требований и спецификаций (12 % проектов).

В [4] подчеркивается, что все современные методологии предполагают возможность изменения требований в ходе разработки. При этом необходимо помнить, что всякое изменение в требованиях связано с изменением стоимости и сроков выполнения проекта и может влиять на качество *программной системы* (ПС). Тем не

менее базовый объем требований должен быть выявлен к началу проекта по созданию ИС.

В [5] утверждается, что от 40 до 60 % всех проблем в проектах по созданию ИС связаны со стадией сбора и формализации требований.

В [6] подчеркивается, что требования являются основополагающими для любого проекта по реализации ИС. Они обязательно должны учитывать потенциальные запросы всех заинтересованных сторон, которые будут выполнять свои профессиональные функции в рамках проектируемой системы. Таким образом, функциональность ИС полностью определяется ее будущими пользователями. Авторы указывают основные причины неудач (неполнота требований, недостаточное привлечение пользователей, нереалистические ожидания) и факторы успеха (вовлечение пользователей, четкая и ясная постановка требований, реалистичные ожидания) в проектах.

Чем раньше выявлены и утверждены функциональные требования, тем больше вероятность выполнения проекта в регламентированные сроки в рамках установленного бюджета и с наибольшей степенью удовлетворения ожиданий заказчика.

В [7] авторы выделяют понятие «хорошие требования», понимая под этим их реализуемость, то есть возможность их программирования. Эти же авторы обращают внимание на то, что ошибки, обусловленные некорректным формированием требований, могут порождать множество проблем соответствующего про-

екта, включая его полный провал. При этом среди основных причин, провоцирующих такие ситуации, преобладают две: плохая подготовленность специалистов со стороны заказчиков (неумение сформулировать функциональные требования) и недостаточная инструментальная база, которая должна обеспечивать процесс работы с требованиями до момента их программной реализации.

Подходы к формализации функциональных требований при проектировании и разработке ИС

Аспекты формализации требований в литературных источниках рассматриваются крайне редко и, как правило, в контексте UML-диаграмм, например [8]. Но такие графические нотации понятны далеко не каждому эксперту от заказчика даже при современном уровне компьютерной грамотности. Чтобы правильно читать специализированные графические интерпретации процессов и технологических особенностей их выполнения в условиях автоматизированной реализации, непосвященных пользователей необходимо обучать. И даже это не гарантирует абсолютного понимания ими содержательного аспекта подобного рода схем. Кроме того, UML-диаграммы нельзя считать панацеей, поскольку с точки зрения программирования они не содержат исчерпывающей информации, необходимой для создания полноценного функционального ПО.

Рассмотрим несколько принципиально различных подходов к понятию «формализация функциональных требований к ИС».

По мнению авторов [9], наличие формализации при управлении большими проектами (включающими большой объем функциональности) является очень ценным качеством, так как позволяет обеспечить его прозрачность.

В угоду современной моде на использование языка UML разработчики функционального ПО практически отказались от такого инструмента, как описание задачи для программирования. Между тем представление задачи в таком виде позволяет описать требования в контексте реально программируемых процедур (задание форматов для вводимых и выводимых данных, определение структур входных и выходных документов, алгоритмов преобразования информации и др.).

В [9] для упрощения процесса разработки функциональных требований предлагается использовать карты воздействий и карты исто-

рий, а также приводится форма таблицы, позволяющая осуществлять трассировку такого рода требований. Показаны две таблицы, которые относятся к организационно-технологическому сопровождению проекта. Первая таблица (деятельность аналитиков и проектировщиков опережает работу программистов на одну-две итерации) описывает роли участников проекта в рамках соответствующих итераций, где под итерацией подразумевается программная реализация определенного требования. Вторая отражает адаптированный бэклог проекта (соответствие функциональных требований, итераций по их реализации и дополнительные характеристики, позволяющие отслеживать динамику программирования требований).

Таким образом, в статье [9] особое внимание уделяется важности постановки задачи, отражающей формализацию функциональных требований, и упорядочению управленческих процедур при выполнении проекта.

Интересный подход к формализации функциональных требований изложен в [10]. Автор подчеркивает, что представители бизнеса, участвующие в процессе формулирования требований, должны уметь изложить их в такой стилистической и семантической форме, чтобы она была понятна как будущим пользователям, так и разработчикам. В качестве такого представления предлагаются две модели управления бизнес-требованиями на примере реализации финансового инструмента «Депозит на расчетном счете»: 1) для обработки бизнес-требования – В2 «Расчет процентов в зависимости от среднемесячной суммы остатка»; 2) для обработки требования пользователя – С5 «Отражение фактических выплат в отчете без учета разбора выписки».

Далее автор предлагает вариант некоторой формализации требований в форме таблиц, в одной из которых описывается алгоритм обработки информации. Такое описание существенно отличается от рекомендуемых UML-диаграмм, содержит в себе элементы формализации и носит частный характер. При большом количестве взаимозависимых требований, часть из которых может предусматривать достаточно сложные алгоритмы, включающие циклические и логические процессы, представленная модель будет непригодна, так как ее текстовое опи-

сание объемно и поэтому трудно воспринимается. Однако для несложных алгоритмов с небольшой долей логических проверок данная модель может быть применима.

Другой подход к формализации требований на основе шаблонов изложен в [6]. По мнению авторов, одним из способов стандартизации языка для формализации требований может быть соответствующий шаблон. Такие шаблоны целесообразно разрабатывать для требований различных типов. В дальнейшем по результатам их практического применения они могут корректироваться как по семантике и стилистике, так и по количеству. Более того, шаблоны, используемые в рамках одного проекта, могут быть распространены и на другие аналогичные или смежные проекты ([6]).

Эти же авторы приводят примеры схем для формулирования требований [6]:

– **Типичное требование**, описывающее возможность (потребность), то есть **<Тип пользователя>** должен иметь возможность **<описание возможности>**;

– **Требование типа <ограничение>**, то есть **<Тип пользователя>** не должен попадать под действие **<соответствующее законодательство>** и **<Система>** должна **<выполняемая функция>** не менее чем **<количество>** **<объект>** функционируя в **<условия эксплуатации>**;

– **Периодическое ограничение**, то есть **<Система>** должна **<выполняемая функция>** **<объект>** каждые **<показатель производительности>** **<единица измерения>**.

Данную схему иллюстрирует рисунок 2.

Рис. 2. Глобальные шаблоны

Fig. 2. Global patterns

Требования к создаваемой ИС также должны быть аккумулированы в форме документа. В [5] отдельная глава посвящена документированию требований, где утверждается, что любое изображение стоит 1 024 слов. По мнению авторов, наиболее подходящими способами представления информации о требованиях с использованием таких элементов могут быть, например, следующие:

- внешние интерфейсы системы (контекстная диаграмма, диаграмма вариантов использования, диаграмма потоков данных, макеты отчетов и др.);
- поток бизнес-процессов (диаграмма потоков, диаграммы действий и др.);
- определения данных и связи объектов данных (диаграмма сущность–связь, диаграмма классов и др.);
- состояния системы и объектов (диаграммы переходов состояния, таблицы состояния, таблица событий и реакций, функциональные требования);
- сложная логика (дерево решений, таблица решений);
- пользовательские интерфейсы (карта диалоговых окон, раскадровки, подробные макеты экранов, описания элементов управления пользовательским интерфейсом и др.);
- описания задач пользователей (пользовательские истории, спецификации сценариев, варианты использования, блок-схемы, диаграммы действий и др.);
- нефункциональные требования (атрибуты качества, ограничения).

Для изложения спецификации требований в [5] предложен шаблон, в который включены общее описание (классы и характеристики пользователей, операционная среда и др.), функциональные требования, требования к данным и т.п.

Следует различать представление функциональных требований для общения с заказчиками и для программистов. Это должно быть отражено и в составе документов к проекту, то есть описание постановки задачи для программиста и для согласования требований с заказчиком – разные документы.

Все перечисленные графические способы представления требований эксперту от заказчика будут ему непонятны (за исключением, возможно, отдельных специалистов). Необходимо показать требования в понятном для эксперта виде.

Очень часто специалисты, говоря о требованиях к функциональному ПО, ссылаются

на [11]. Но, несмотря на то, что в данном документе представлены содержание и характеристики, предоставляющие возможности для составления качественных спецификаций к ПО, эта методика не содержит в себе какой-либо конкретный метод, терминологическую базу или инструментарий для подготовки SRS (Software Requirements Specification) [11]. В этом же документе говорится об языках спецификаций требований, которые позволят нивелировать неоднозначности, характерные для естественного языка. При этом отмечаются длительность их изучения и остающаяся непонятность для неподготовленных пользователей.

Итак, краткий анализ нескольких интересных источников по означенной тематике выявил следующие особенности.

1. Изучению проблем формализации функциональных требований к ИС посвящено очень мало публикаций. Работы зарубежных авторов присутствуют на российском рынке в основном в виде монографий, причем с достаточно глубоким погружением в тему. Результаты исследований российских авторов опубликованы в основном в научной периодической печати.

2. Принципиальные идеи представления и описания требований к ИС (включая функциональные требования) исходят от зарубежных авторов. Эти идеи изложены ими в фундаментальных трудах соответствующей тематики.

3. Отдельно рассматриваются требования к функциональному ПО и функциональные требования к ИС. Однако, поскольку и функциональное ПО, и функциональная часть ИС в конечном виде – это собственно программа, можно считать эти две разновидности требований эквивалентными.

4. Существуют несколько уровней работы с требованиями:

- выявление требований (сбор первичной информации о функциональности, которая должна быть реализована);
- структурное преобразование первичной информации в формальные семантические (текстовые) конструкции, предназначенные для обсуждения с заказчиком;
- формирование функциональной составляющей ИТ-проекта по созданию функционального ПО;
- составление спецификаций (описание постановок задач), являющихся основой для программирования.

5. Во всех публикациях, как фундаментального характера, так и в виде научных статей,

основное внимание уделяется управлению требованиями (например [12]), а не их корректным формулировкам. Объяснить это можно тем, что в связи с огромным разнообразием предметных областей и решаемых в их рамках задач очень сложно найти какие-либо унифицированные математические или логические конструкции, подходящие если не для всех из них, то, по крайней мере, для большинства. Использование пользовательских историй или описаний прецедентов может быть одним из вариантов решения проблемы, но тоже не универсальным.

6. Возможность формализации функциональных требований разные авторы рассматривают по-разному в рамках какого-либо одного их перечисленных уровней. Таким образом, в настоящее время отсутствует некоторый унифицированный подход к данному вопросу.

Поскольку существует следующая дифференциация: система → подсистема (модуль) → задача, то по сути требования к системе трансформируются в требования к решаемым задачам.

Ни один проект по созданию функционального ПО не может существовать сам по себе, то есть отдельно от соответствующей предметной области. Он всегда является прикладным. И поэтому первичные условия для него – объем реализуемой в проекте функциональности, то есть количество решаемых задач, и сложность информационных взаимодействий между задачами, а также смежными ИС на внутрикорпоративном уровне и во внешней среде.

В свою очередь, это определяет требования к содержанию собственно проекта, которые представляют собой необходимые и достаточные условия для его реализации. К ним, в частности, относятся:

- подбор команды для выполнения проекта;
- составление плана работ по реализации проекта;
- распределение работ по проекту среди участников команды;
- контроль за выполнением работ;
- регулирование времени выполнения проекта (при необходимости);
- анализ выполнения хода работ по проекту;
- обеспечение соответствующей технической базы;
- обеспечение требуемыми ИТ, включая инструментальные средства программирования и ведения БД.

В [13] был предложен подход, заключающийся в том, чтобы начиная со второго уровня выражать функциональные требования через названия задач по функциям управления. Это позволит сократить разрыв между требованиями как таковыми и их алгоритмической сущностью, что является важнейшим условием программной реализации, а значит, и более упорядоченного управления ими в рамках выполняемого проекта. Более того, такие термины, как *учет, планирование, контроль, анализ, регулирование*, будут понятны и представителям (например, бизнес-среды), и разработчикам функционального ПО.

Любой ИТ-проект, имеющий на выходе программный продукт в виде ИС или ее отдельного модуля, основан на двух принципиальных положениях:

- все подлежащие реализации в рамках проекта функциональные требования можно стилистически отразить через функции управления (планирование, учет, контроль, анализ, прогнозирование, принятие решений, регулирование) или типовые процедуры обработки информации (поиск, передача, представление);
- проект будет успешным, если функциональные требования к его содержанию зафиксированы однозначно и утверждены ключевыми заинтересованными сторонами.

Реализация каждого алгоритма из функциональных требований к ИС базируется на одном из трех атомарных информационно-технологических действий:

- ввод данных, включая редактирование;
- вычисление, то есть работа с БД;
- вывод данных.

Пример ввода информации по шаблону

Задачи учетного характера с алгоритмической точки зрения сводятся к формированию (или редактированию) соответствующих полей в БД, то есть вычислительная основа задачи ориентирована на ввод информации (отдельные поля в существующих файлах или вновь создаваемый файл). В качестве шаблона можно рассмотреть универсальную структурную форму в виде простейшей таблицы, состоящей всего из двух столбцов (рис. 3). При необходимости второй столбец может быть размножен (например, если для одного студента надо указать оценки по всем экзаменам во время сессии). Вопрос о выборе формы для ввода (таб-

Название показателя	Значение показателя	...	Значение показателя
1	2	...	m
Показатель 1	Значение 1.1	...	Значение 1.1
Показатель 2	Значение 2.1	...	Значение 2.1
...	...	...	...
Показатель n	Значение n.1	...	Значение n.m

Рис. 3. Шаблон простейшей формы документа для ввода информации

Fig. 3. The simplest template of a data input document

лица или анкета, то есть таблица, состоящая всего из двух столбцов) является одним из аспектов проектирования и зависит от количественного и качественного состава вводимой информации.

При разнесении отдельных показателей таблицы в пределах плоскости экрана можно получить документ любой произвольной формы. При этом технологически сущность заполнения полей будет аналогична табличной.

Присутствие в спецификации функциональных требований описаний на основе атомарных действий послужит основой для расчета трудоемкости проекта. При этом следует учитывать, в частности, такие характеристики приведенной формы:

- количество вводимых показателей;
- количество показателей, предусматривающих ручной ввод информации в произвольной символьной форме;
- количество показателей, значения кото-

рых выбираются из заданного списка (словаря, справочника, классификатора);

- количество показателей, значения которых вводятся в соответствии с некоторым определенным шаблоном (маской);

- количество сообщений системы, предупреждающих или блокирующих ввод неправильных значений.

Таким образом, в качестве варианта для формализации функциональных требований может быть предложена следующая последовательность.

Этап 1: выявление требований с учетом потенциальной алгоритмической составляющей при их программной реализации.

Этап 2: структурное преобразование первичной информации в формальные семантические (текстовые) конструкции (использовать в названиях задач, которые подлежат последующему программированию, классические функции управления или технологические процедуры обработки информации).

Этап 3: формирование функциональной составляющей проекта по созданию ИС на основе результатов второго этапа.

Этап 4: составление спецификаций (описание постановок задач) с использованием атомарных действий, лежащих в основе составления программ.

Предложенный подход, вероятно, нельзя считать полностью унифицированным, но он может рассматриваться как некоторая стандартизация для определенного типа задач, которые в рамках проектирования и разработки ИС подлежат программированию.

Литература

1. Бизнес-проект «Качели». URL: <http://www.drive2.ru/b/288230376151782390/> (дата обращения: 22.01.2019).
2. Колоденкова А.Е. Актуальные проблемы современной программной инженерии: концептуальное проектирование и жизнеспособность программного проекта // Вестн. УГАТУ. 2011. Т. 15. № 2. С. 8–21.
3. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. М.: Вильямс, 2002. 448 с.
4. Гусятников В.Н., Безруков А.И. Стандартизация и разработка программных систем. М.: Финансы и статистика, 2010. 284 с.
5. Вигерс К., Битти Дж. Разработка требований к программному обеспечению; [пер. с англ.]. М.–СПб, 2014. 716 с.
6. Халл Э., Кен Дж., Дик Дж. Разработка и управление требованиями. UK: Telelogic, 2005. 240 с.
7. Гвоздев В.Е., Ровнейко Н.И. Численное моделирование процессов формирования требований к программному продукту // Вестн. УГАТУ. 2012. Т. 16. № 8. С. 52–60.
8. Яловой И. Анализ требований и управление изменениями программных проектов // Инженерн. вестн. Дона. 2008. № 11. С. 32–42.

9. Карпов В.В., Карпов А.В. Особенности применения современных методов разработки программного обеспечения защищенных автоматизированных систем // Программные продукты и системы. 2016. № 1. С. 5–12. DOI: 10.15827/0236-235X.113.005-012.
10. Кравченко Т.К. Управление требованиями при реализации ИТ-проектов // Бизнес-информатика. 2013. № 3. С. 63–71.
11. IEEE 830-1998. URL: http://kspt.icc.spbstu.ru/media/files/2009/course/se/IEEE-830-1998_RU.doc (дата обращения: 21.01.2019).
12. Сулейманова А.М., Яковлев Н.Н. Семантическая аннотация и многоаспектная модель данных в управлении требованиями // Программные продукты и системы. 2011. № 2. С. 45–48.
13. Гутгарц Р.Д. Роль бизнес-процессов в формировании требований к ERP-системе // Проблемы теории и практики управления. 2015. № 1. С. 118–127.

Software & Systems
DOI: 10.15827/0236-235X.127.349-357

Received 30.01.19
2019, vol. 32, no. 3, pp. 349–357

On the formalization of functional requirements in information system projects

R.D. Gutgarts¹, Dr.Sc. (Economics), Professor, gutgarc@gmail.com

E.I. Provilkov¹, Postgraduate Student, provilkoff@gmail.com

¹ Irkutsk National Research Technical University, Irkutsk, 664074, Russian Federation

Abstract. There are a lot of scientific and applied research devoted to discussion of problems in IT project management. Then mostly focus on the financial aspects and duration of a project. However, functional aspects are often overlooked. This might be explained by the fact that such indicators as money and time can be calculated using appropriate methods and algorithms taking into account various risks. Meanwhile, a reasonable numerical equivalent to determine project functionality still does not exist.

When considering such specialized IT project as the design and development of an information system (or its separate module), the implemented functionality is a fundamental factor which affects all other project indicators. However, fundamental literature and periodical scientific publications pay insufficient attention to functionality as a semantic item for various reasons. Scientists and specialists are mostly interested in managing information systems requirements including functional ones. The formalization of functional requirements is a subject for discussion in a scientific community and is considered in various aspects. However, there are no standardized or unified solutions.

The paper considers the issues related to functional requirements, features of their initial formulation, presentation for discussion with a customer and with formalization for software implementation in a project.

The authors briefly analyze approaches to the formalization of requirements and propose an approach to the formalization of functional requirements, which can be applied to certain types of tasks that are included in the software that represents an information system as a software product. This may be the first step to create prerequisites for the development of an algorithmic component. Thus, for a more correct calculation of project complexity, and for more accurate planning of its financial and time costs.

Keywords: IT project, information systems design, information systems functional requirements, functional requirements formulating problem, functional requirements formalization, functional requirements unification.

References

1. *Kacheli Business Project*. Available at: <http://www.drive2.ru/b/288230376151782390/> (accessed January 22, 2019).
2. Kolodenkova A.E. Relevant problems of modern software engineering: conceptual design and viability of a software project. *Bulletin of USATU*. 2011, vol. 15, no. 2, pp. 8–21 (in Russ.).
3. Leffingwell D., Widrig D. *Managing Software Requirements. A Unified Approach*. Addison-Wesley Prof. Publ., 1999, 528 p. [Russ. ed.: Moscow, Vilyams Publ., 2002, 448 p.].
4. Gusyatnikov V.N., Bezrukov A.I. *Standardization and Development of Software Systems*. Moscow, Finansy i statistika, INFRA-M Publ., 2010, 284 p.

5. Wiegers K., Beatty J. *Software Requirements*. 3rd ed., Microsoft Press, 2013, 672 p. [Russ. ed.: Moscow, BHV Publ., 2014, 716 p.].
6. Hull E., Jackson K., Dick J. *Requirements Engineering*. 2nd ed., UK, Springer Publ., 2005, 198 p. [Russ. ed.: Telelogic Publ., 2005, 240 p.].
7. Gvozdev V.E., Rovneyko N.I. Numerical modeling of software requirement formation. *Bulletin of USATU*. 2012, vol. 16, no. 8, pp. 52–60 (in Russ.).
8. Yalovoy I. Requirements analysis and change management for software projects. *Engineering J. of Don*. 2008, no. 11, pp. 32–42 (in Russ.).
9. Karpov V.V., Karpov A.V. Application features of modern methods of software development of secure automated systems. *Software & Systems*. 2016, no. 1, pp. 5–12 (in Russ.). DOI: 10.15827/0236-235X.113.005-012.
10. Kravchenko T.K. Requirement management in the implementation of IT projects. *Business Informatics*. 2013, no. 3, pp. 63–71 (in Russ.).
11. *IEEE 830-1998*. Available at: http://kspt.icc.spbstu.ru/media/files/2009/course/se/IEEE-830-1998_RU.doc (accessed January 21, 2019).
12. Suleymanova A.M., Yakovlev N.N. Semantic annotation and multidimensional data model in requirements management. *Software & Systems*. 2011, no. 2, pp. 45–48 (in Russ.).
13. Gutgarts R.D. The role of business processes in forming requirements for the ERP system. *Problems of Management Theory and Practice*. 2015, no. 1, pp. 118–127 (in Russ.).

Для цитирования

Гутгарц Р.Д., Провилков Е.И. О формализации функциональных требований в проектах по созданию информационных систем // Программные продукты и системы. 2019. Т. 32. № 3. С. 349–357. DOI: 10.15827/0236-235X.127.349-357.

For citation

Gutgarts R.D., Provilkov E.I. On the formalization of functional requirements in information system projects. *Software & Systems*. 2019, vol. 32, no. 3, pp. 349–357 (in Russ.). DOI: 10.15827/0236-235X.127.349-357.