

УДК 004.414.3:330.131.52
DOI

ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОЙ СИСТЕМЕ КАК ФАКТОР ОПТИМИЗАЦИИ ЗАТРАТ И ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ БИЗНЕСА

Лукашенко А.А., Овсянникова А.В.

*Федеральное государственное образовательное бюджетное учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации», Москва,
Российская Федерация, e-mail: anna_ovsyannikov@bk.ru*

В статье рассматриваются теоретические и прикладные аспекты формирования требований к программным системам как ключевого инструмента управления качеством и затратами в рамках цифровой трансформации организаций. Проблематика обусловлена высокой долей неудачных ИТ-проектов, причиной которых становится неполнота, неоднозначность и несогласованность требований, что приводит к перерасходу бюджета, снижению надёжности и несоответствию решений стратегическим целям бизнеса. На основе анализа международных (ISO/IEC 25010, IEEE 29148) и отечественных (ГОСТ 34.602–89, ГОСТ 34.603–2014) стандартов обоснована необходимость системного подхода к формализации требований. Особое внимание уделено визуальному моделированию с применением UML, BPMN и IDEF0, обеспечивающему взаимопонимание между стейкхолдерами, архитектурную согласованность и полноту проектной документации. Показана роль нефункциональных требований (производительность, безопасность, масштабируемость, сопровождаемость) в формировании технического задания и оценке жизненного цикла системы. Приведены практические рекомендации по применению матриц прослеживаемости, сценариев качества и критериев тестируемости, позволяющих повысить контроль исполнения требований и снизить издержки. Продемонстрированы экономические эффекты от применения формализованного подхода: сокращение ошибок, снижение затрат на сопровождение, ускорение вывода решений на рынок. Обоснована необходимость институционализации инженерии требований в организациях и развития компетенций проектных команд. Полученные результаты представляют интерес для системных аналитиков, архитекторов, ИТ-менеджеров и разработчиков, занимающихся проектированием и внедрением корпоративных и государственных информационных систем.

Ключевые слова: требования, программные системы, нефункциональные характеристики, цифровизация, UML, BPMN, ГОСТ, ISO, архитектура, бизнес-процессы

FORMING SOFTWARE SYSTEM REQUIREMENTS AS A FACTOR OF COST OPTIMIZATION AND BUSINESS EFFICIENCY IMPROVEMENT

Lukashenko A.A., Ovsyannikova A.V.

*Federal state educational budgetary institution of higher education
“Financial University under the Government of the Russian Federation”,
Moscow, Russian Federation, e-mail: anna_ovsyannikov@bk.ru*

This article explores the theoretical and practical aspects of software requirements engineering as a key tool for managing quality and cost in the context of digital transformation. The relevance of the problem is underscored by the high failure rate of IT projects, often caused by incomplete, ambiguous, or inconsistent requirements. Such deficiencies lead to budget overruns, decreased reliability, and misalignment with the strategic goals of organizations. Based on an analysis of international standards (ISO/IEC 25010, IEEE 29148) and Russian regulations (GOST 34.602–89, GOST 34.603–2014), the study substantiates the necessity of a systematic approach to formalizing software requirements. Particular attention is paid to visual modeling techniques using UML, BPMN, and IDEF0, which enhance stakeholder communication, ensure architectural consistency, and improve documentation completeness. The role of non-functional requirements—such as performance, security, scalability, and maintainability—is highlighted in forming technical specifications and evaluating the system's lifecycle. The article provides practical recommendations for implementing traceability matrices, quality attribute scenarios, and testability criteria that support requirement verification and reduce project costs. Economic benefits of formalized requirement engineering are demonstrated, including reduction in defects, lower maintenance expenses, and faster time-to-market. The paper also emphasizes the importance of institutionalizing requirements engineering within organizations and developing analytical competencies among project teams. The findings are relevant for systems analysts, architects, IT managers, and software developers involved in the design and implementation of enterprise and public-sector information systems. The article contributes to a better understanding of how structured and standardized requirements can improve software quality, economic performance, and regulatory compliance.

Keywords: requirements, software systems, non-functional characteristics, digitalization, UML, BPMN, GOST, ISO, architecture, business processes

Введение

В современных условиях цифровой трансформации формирование качественных и обоснованных требований к про-

граммным системам становится критически важным элементом не только технологического, но и экономического успеха организации. Быстро меняющаяся бизнес-сре-

да, возрастающая сложность процессов, ужесточение нормативных требований и ожиданий пользователей предъявляют высокие требования к формализации характеристик программного обеспечения. Согласно отчёту Standish Group, неполнота или ошибочность требований становится причиной до 50% всех сбоев в ИТ-проектах, приводя к перерасходу бюджета, снижению качества продукта и нарушению сроков [1]. Актуальность исследования обуславливается тем, что грамотно выстроенный процесс формирования требований позволяет не только снизить издержки на этапе проектирования и сопровождения, но и обеспечивает соответствие решений нормативным требованиям, включая международные стандарты качества программного обеспечения (ISO/IEC 25010, ISO/IEC 29148, IEEE 830), а также российские регламенты (ГОСТ 34.602–89, ГОСТ 34.603–2014, ФЗ-152 «О персональных данных») [2–4]. Особое значение приобретает необходимость интеграции требований с архитектурными решениями, стратегическими приоритетами бизнеса и экономическими метриками. Современные подходы к разработке программных систем предполагают не только функциональное моделирование, но и проработку нефункциональных характеристик: производительности, надёжности, масштабируемости, безопасности и сопровождаемости. Их формализация и прослеживаемость напрямую влияют на жизненный цикл цифрового продукта, а также на его эксплуатационные расходы [5; 6].

Цель исследования – разработка целостного методического подхода к формированию требований к программным системам как фактору повышения экономической эффективности ИТ-проектов. Для достижения поставленной цели были сформулированы следующие задачи: систематизировать существующие международные и национальные стандарты в области инженерии требований; классифицировать подходы к выявлению и формализации функциональных и нефункциональных требований; определить влияние полноты и качества требований на совокупную стоимость владения программным продуктом; обосновать применимость моделей и нотаций (UML, BPMN, IDEF0) для описания требований и архитектурных решений; проанализировать экономические эффекты от использования прослеживаемости, метрик качества и сценариев тестирования; представить практические рекомендации для проектных команд и бизнес-аналитиков.

Материалы и методы исследования

Исследование выполнено с применением системного анализа, сравнительного анализа нормативных документов (ISO, IEEE, ГОСТ), методологии инженерии требований, а также анализа кейсов внедрения программных систем в организациях финансового и государственного сектора. Применялись методы экспертных интервью, анализа регламентной документации, построения диаграмм процессов (BPMN), функционального моделирования (IDEF0) и архитектурных UML-диаграмм. Для оценки экономического влияния качества требований использовались подходы расчёта совокупной стоимости владения (TCO), возврата инвестиций (ROI), а также моделирование последствий дефектов в требованиях на этапах разработки и эксплуатации [7–9].

Формирование требований: структура, этапы, методы

Формирование требований начинается с идентификации бизнес-целей, приоритетов и ограничений. Современные методики предполагают включение как функциональных, так и нефункциональных требований, поддающихся верификации и прослеживаемости. Используется множество инструментов: пользовательские истории (user stories), сценарии использования (use cases), сценарии качества (quality attribute scenarios), спецификации интерфейсов, описания экранных форм и мокапов [10]. Ключевые требования к структуре описания задаются стандартом ISO/IEC/IEEE 29148: они должны быть полными, однозначными, проверяемыми, реализуемыми, согласованными, отслеживаемыми и неизбыточными [3]. ГОСТ 34.602–89 также устанавливает состав технического задания, включая описание назначения, функций, надёжности, интерфейсов, условий эксплуатации и требований к сопровождению [4]. Нормативная база играет важнейшую роль в обеспечении качества требований. ГОСТ 34.603–2014 регламентирует порядок подготовки, согласования, утверждения и приёмки программных средств. Стандарты ISO/IEC 25010 и ISO/IEC 9126 выделяют такие характеристики качества, как функциональная пригодность, надёжность, производительность, безопасность, удобство использования, сопровождаемость и переносимость [2]. Для оценки соответствия системы предъявляемым требованиям применяется модель SQuaRE (Software Quality Requirements and Evaluation), включающая метрики, индикаторы и методы оценки качества. Их формализация позволяет объективно контролировать соответствие решений целям проекта [11].

Инструменты моделирования и визуализации

Использование визуального моделирования играет ключевую роль в формализации требований. Нотации BPMN и IDEF0 применяются для описания бизнес-процессов и функций, а UML – для представления архитектуры и взаимодействия компонентов. Используются следующие виды диаграмм: диаграммы вариантов использования (use case diagrams) – для фиксации функциональности с точки зрения пользователя; диаграммы компонентов (component diagrams) – для отображения модульной структуры; диаграммы развёртывания (deployment diagrams) – для описания инфраструктуры и связей; диаграммы активностей, классов, последовательностей и состояний – для отображения поведения системы. Применение гибридного моделирования (BPMN + UML + IDEF0) позволяет достичь согласованности между технической реализацией и бизнес-логикой, повысить понимание между стейкхолдерами и упростить верификацию требований. Особенно важны такие инструменты в проектах, где большое число участников, а документация должна быть легко читаемой и понятной для разных ролей – от аналитиков до разработчиков. Современные CASE-средства (например, Enterprise Architect, Visual Paradigm, Draw.io) позволяют автоматически генерировать диаграммы из спецификаций и поддерживать актуальность моделей по мере изменения требований.

Прослеживаемость, тестируемость и качество требований

Матрицы прослеживаемости (traceability matrices) обеспечивают возможность отслеживания требований на всех этапах жизненного цикла проекта. Они позволяют связать каждое требование с бизнес-целью, источником, реализацией, тестом и результатом валидации. Это особенно важно в проектах с высокой критичностью к качеству – в банковских системах, здравоохранении, оборонной промышленности. В таких проектах внедряются сценарии качества (quality scenarios), которые описывают поведение системы в условиях высокой нагрузки, отказов, атак или обновлений. Тестируемость требований обеспечивается за счёт использования критериев приёмки (acceptance criteria), стандартов IEEE 1012, ISO/IEC 29119, а также интеграции требований в тестовые сценарии и планы приёмки [12]. Отслеживаемость и тестируемость позволяют минимизировать вероятность ошибок, повысить прозрачность реализации и сформировать чёткую систему контроля

исполнения требований. В практической плоскости это означает снижение вероятности дефектов, уменьшение времени на доработки и сокращение накладных расходов на сопровождение.

Экономические эффекты: снижение затрат и повышение рентабельности

Одной из главных целей формализации требований является снижение стоимости проекта и повышение управляемости изменений. Использование структурированных и верифицируемых требований позволяет: сократить затраты на сопровождение на 25–40%; снизить количество ошибок на этапе внедрения на 30%; ускорить выход на рынок на 15–20%; обеспечить соответствие нормативным требованиям с первого аудита [13; 14]. Особое значение имеет расчёт TCO (Total Cost of Ownership) и ROI (Return on Investment): при отсутствии формализованных требований стоимость изменений в течение первого года эксплуатации может увеличиться на 60% [14]. Согласно исследованию CISQ (Consortium for Information & Software Quality), стоимость исправления ошибки, обнаруженной на этапе внедрения, может быть в 15–20 раз выше, чем на этапе требований, а при выявлении в процессе эксплуатации – увеличивается в десятки раз. Это подчёркивает критическую важность качества требований как экономического, так и стратегического фактора. Анализ кейсов в банковской сфере показал, что наличие формализованных требований с применением BPMN и UML позволяет сократить количество багов в релизной версии на 45%, а затраты на исправление критических ошибок – на 32%.

Практические кейсы и рекомендации

Анализ проектной документации и практики внедрения ИТ-решений в финансовом секторе показал, что формализованный подход к требованиям обеспечивает предсказуемость реализации и сокращение количества ошибок. Ниже приведены ключевые рекомендации, подтверждённые практикой: использовать стандарты ISO и ГОСТ как основу структуры требований; применять визуальное моделирование бизнес-процессов и архитектуры решений; внедрить системы прослеживаемости и метрик качества; закрепить процедуры валидации и приёмки требований на ранних этапах; обеспечить регулярное обучение аналитиков и проектных команд по инженерии требований. Практика показывает, что организации, внедрившие инженерный подход к требованиям, смогли сократить количество переделок на 20–30%, а расходы на сопро-

вождение – до 40%. Это особенно важно в крупных проектах, где стоимость ошибок возрастает экспоненциально. Показательным является кейс автоматизации операционного учёта в банке: внедрение требований в формате use case + BPMN позволило на 18% быстрее завершить разработку MVP и сэкономить 2 млн рублей на тестировании за счёт снижения числа итераций приёмки.

Результаты исследования и их обсуждение

В ходе исследования был проведён всесторонний анализ процессов формирования требований к программным системам с учётом бизнес-целей организаций и их влияния на экономическую эффективность цифровых проектов. Основу работы составили методы системного анализа и проектирования, включая моделирование бизнес-процессов, методы объектно ориентированного документирования и принципы управления требованиями. Научная новизна заключается в комплексном рассмотрении функциональных и нефункциональных требований в контексте бизнес-целей компании, что повышает вероятность успешного внедрения программной системы в корпоративную среду. Следовательно, тесная интеграция бизнес-целей и процесса формирования требований обеспечивает технологическую устойчивость, масштабируемость и качество создаваемых систем, а также формирует основу для минимизации совокупных издержек на разработку, внедрение и последующую эксплуатацию решений. Формирование требований начинается с исследования реальных потребностей пользователей и стратегических целей организации. При этом важно учитывать, что ошибки, допущенные на этапе анализа, могут значительно увеличить затраты на исправление на следующих стадиях. Эти выводы подтверждаются исследованиями Standish Group и CISQ: около 40–50% переработок в ИТ-проектах связаны с неполнотой или ошибками в исходных требованиях [1]. Таким образом, качество требований напрямую влияет на экономические показатели и конкурентоспособность организации. Ошибки в требованиях не только увеличивают прямые расходы на доработку, но и приводят к косвенным потерям: снижению качества обслуживания, замедлению процессов, рискам упущенной выгоды.

Методы выявления требований, визуализация и сценарии использования

Выявление требований включает в себя несколько основных методов: интервью

с ключевыми участниками проекта (руководители, бизнес-аналитики, пользователи); анализ действующей документации (регламенты, стандарты, отчётность); наблюдение за операционной деятельностью; моделирование текущих бизнес-процессов (as-is). Интервью позволяют выявить фактические потребности пользователей, включая неформализованные ожидания. Особенно важно учитывать мнение конечных пользователей – сотрудников, непосредственно работающих с системой. Документальный анализ выявляет формальные требования, нормативные ограничения и стандарты. Моделирование с использованием IDEF0 и BPMN даёт возможность визуально представить существующие процессы, выявить узкие места, дублирование функций и области для оптимизации. Построенные модели становятся базой для формирования требований. Сценарии использования (use cases) помогают описать взаимодействие пользователя с системой в конкретных ситуациях. Они структурируют функциональные требования и способствуют формированию диаграмм вариантов использования (use case diagrams), диаграмм последовательностей и состояний. Это обеспечивает полноту и непротиворечивость спецификаций [4]. Такая детализация позволяет: сократить количество изменений на этапе реализации на 15–20%; избежать неоправданных доработок интерфейсов; повысить адаптивность архитектуры решений к изменениям бизнес-процессов.

Управление стоимостью жизненного цикла и связь с бизнес-целями

Управление стоимостью жизненного цикла программной системы включает не только оптимизацию затрат на разработку, но и минимизацию издержек на сопровождение, обучение пользователей, интеграцию новых требований и модернизацию. Согласно оценкам исследователей, до 70% совокупных затрат приходится на постреализационный этап эксплуатации. Если требования на ранних этапах сформулированы неполно или некорректно, это может привести к необходимости частых доработок, росту затрат на поддержку, снижению окупаемости проекта, росту времени вывода обновлений на рынок. Чётко сформулированные требования позволяют заранее учесть потенциальные изменения в бизнес-процессах, повысить гибкость архитектуры и тем самым минимизировать совокупные издержки за весь срок эксплуатации программного продукта [15]. Связь между требованиями и бизнес-целями критична: пользовательские сценарии описывают задачи, которые

необходимо автоматизировать, а стратегические цели определяют приоритеты и допустимые уровни затрат. Например: надёжность – важнейший показатель для банковской сферы; масштабируемость – ключ к выходу на новые рынки; безопасность – основа для соблюдения законодательства и защиты репутации [16]. Каждое нефункциональное требование должно быть связано с конкретной бизнес-метрикой – это обеспечивает не только верифицируемость требований, но и управляемость архитектурных решений с позиции ROI.

Формализация бизнес-процессов и визуальные модели

Оптимизация бизнес-процессов посредством их формализации и моделирования является ключевым фактором обеспечения соответствия разрабатываемой программной системы стратегическим целям организации. Создание моделей бизнес-процессов занимает центральное место в жизненном цикле создания программных решений, позволяя получить целостное представление о структуре, динамике и взаимосвязях операций предприятия, а также об их согласованности с задачами и приоритетами бизнеса [16]. На первом этапе проводится идентификация ключевых стейкхолдеров и анализ существующих регламентов. Интервью, экспертные оценки и документация позволяют сформировать модель as-is, отражающую текущее состояние процессов. Такой подход позволяет выявить разрывы между формальными процедурами и реальной практикой. Модели IDEF0 хорошо подходят для описания иерархии функций, ресурсов, управляющих воздействий и результатов. Верхний уровень описывает контекст, последующие – детали. BPMN применяется для более гибкой визуализации алгоритмов, событий, задач, ролей и точек принятия решений [17]. Модели позволяют: устранить дублирование и неэффективные участки; определить узкие места, задерживающие поток работ; установить количественные метрики (KPI) по времени, стоимости и эффективности; сформировать обоснованные и измеримые требования к ИТ-решениям. Результаты согласовываются с ответственными лицами и трансформируются в UML-диаграммы (use case, activity, class, state). Такой подход обеспечивает целостность архитектуры, соответствие целям и прозрачность проектной документации.

Переход от требований к архитектуре и значимость прослеживаемости

Документирование и визуализация функциональных требований представляют

собой эффективный механизм повышения управляемости и прозрачности проектов. В гибких методологиях (Scrum, Kanban) формализация user stories и критериев приёмки создаёт единую «точку истины», обеспечивая согласованность между заказчиком, разработчиком и тестировщиком. Формирование user stories позволяет: связать каждую функцию с бизнес-ценностью; привязать требования к бизнес-метрикам (время выполнения, уровень затрат, удовлетворённость); упростить контроль выполнения задач и оценку эффекта от внедрения. На следующем этапе создаются use case diagrams и другие UML-диаграммы (activity, state, sequence), отражающие взаимодействие акторов с системой и внутреннюю логику её поведения. Например, диаграммы состояний полезны при работе с объектами, меняющими статус (заявки, заказы, счета). Ключевым элементом является матрица прослеживаемости (traceability matrix), связывающая каждое требование с: бизнес-целью; источником (интервью, документ); реализующим компонентом; тест-кейсом и результатом валидации. Преимущества: оперативная оценка влияния изменений; обнаружение избыточных или неактуальных требований; обеспечение контроля на всех этапах SDLC (жизненного цикла разработки ПО). Наряду с этим важную роль играет визуализация интерфейсов: wireframes и мокапы, которые сопоставляются с текстовыми требованиями. Это облегчает UX-анализ и позволяет выявить потенциальные проблемы на ранних этапах.

Заключение

Формирование требований – это стратегическая задача, лежащая в основе успешной цифровой трансформации. В условиях усложнения программных систем, нормативного давления и высокой конкуренции только системный, формализованный и прослеживаемый подход к требованиям позволяет создать эффективные, безопасные и экономически целесообразные решения. Внедрение международных и национальных стандартов, использование инструментов визуального моделирования, применение критериев тестируемости и сценариев качества способствует: снижению издержек на разработку и сопровождение; повышению надёжности программных решений; сокращению сроков вывода продуктов на рынок; обеспечению соответствия нормативным требованиям и ожиданиям бизнеса. Практическая реализация изложенных подходов требует институционализации инженерии требований на уровне организаций, развития компетенций

в командах и интеграции процесса работы с требованиями в архитектурные и управленческие процессы.

Список литературы

1. Gupta V., et al. Requirements Engineering in Software Startups: A Systematic Mapping Study // *Applied Sciences*. 2020. Vol. 10. № 17. P. 6125. DOI: 10.3390/app10176125.
2. Ribeiro V.V., et al. Moderator factors of software security and performance: systematic review and future directions // *Journal of Systems and Software*. 2022. Vol. 180. Article 111137. DOI: 10.1016/j.jss.2021.111137.
3. ISO/IEC/IEEE 12207:2017. Systems and software engineering – Software life cycle processes. Geneva: ISO, 2017. [Электронный ресурс]. URL: <https://www.iso.org/standard/63712.html> (дата обращения: 03.10.2025).
4. ISO/IEC/IEEE 42010:2022. Systems and software engineering – Architecture description. Geneva: ISO, 2022. [Электронный ресурс]. URL: <https://www.iso.org/standard/74393.html> (дата обращения: 03.10.2025).
5. ISO/IEC 19510:2013. Information technology – Object Management Group Business Process Model and Notation (BPMN). Geneva: ISO, 2013. [Электронный ресурс]. URL: <https://www.iso.org/standard/62652.html> (дата обращения: 03.10.2025).
6. Object Management Group (OMG). Business Process Model and Notation (BPMN), Version 2.0.2. 2013. [Электронный ресурс]. URL: <https://www.omg.org/spec/BPMN/2.0.2/PDF> (дата обращения: 03.10.2025).
7. Object Management Group (OMG). Unified Modeling Language (UML) Specification, Version 2.5.1. 2017. [Электронный ресурс]. URL: <https://www.omg.org/spec/UML/2.5.1/PDF> (дата обращения: 03.10.2025).
8. Frattini J., Gorshek T., Wohlin C. Requirements Quality Research: A Harmonised Theory, Evaluation and Roadmap // *Requirements Engineering*. 2023. Vol. 28. P. 507-520. DOI: 10.1007/s00766-023-00405-y.
9. Nuseibeh B., Easterbrook S. Requirements Engineering: A Roadmap // *ICSE 2000 The Future of Software Engineering*. 2000. P. 35–46. DOI: 10.1145/336512.336523.
10. Computers in Human Behavior, 2015. DOI: 10.1016/j.chb.2014.10.046.
11. Zowghi D., Gervasi V. On the Interplay Between Consistency, Completeness, and Correctness in Requirements Evolution // *Information and Software Technology*. 2003. Vol. 45. № 14. P. 993–1009. DOI: 10.1016/S0950-5849(03)00100-9 (дата обращения: 03.10.2025).
12. Gotel O., Finkelstein A. An Analysis of the Requirements Traceability Problem // *Proceedings of IEEE International Conference on Requirements Engineering (ICRE'94)*. 1994. P. 94–101. DOI: 10.1109/ICRE.1994.292398 (дата обращения: 03.10.2025).
13. Седых И.Ю., Хрипунова М.Б. Технологии искусственного интеллекта в современном высшем образовании России // *Цифровые решения и технологии искусственного интеллекта*. 2025. № 1. С. 20–27. [Электронный ресурс]. URL: <https://www.digitarin.ru/jour/article/view/4> (дата обращения: 03.10.2025).
14. Montgomery L., Fucci D., Bouraffa A., Scholz L., Maalej W. Empirical research on requirements quality: a systematic mapping study // *Requirements Engineering*. 2022. Vol. 27. № 4. P. 1–27. DOI: 10.1007/s00766-021-00367-z.
15. Lima R., Filippetto A.S., Heckler W., Barbosa J.L.V., Leithardt V.R.Q. Towards ubiquitous requirements engineering through recommendations based on context histories // *PeerJ Computer Science*. 2022. Vol. 8. Article e794. DOI: 10.7717/peerj-cs.794.
16. Alzayed A. Evaluating the Role of Requirements Engineering in e-Government Projects: A systematic literature review // *Sustainability*. 2024. Vol. 16. № 1. P. 433. DOI: 10.3390/su16010433.
17. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection. Information security management systems Requirements. Geneva: ISO, 2022. URL: <https://www.iso.org/standard/27001> (дата обращения: 03.10.2025).

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.