

Проблемы и перспективы сбора и трассировки требований в проектах развития

Challenges and Prospects of Requirements Collection and Tracing in Development Projects

DOI: 10.12737/2587-6279-2025-14-4-28-41

Получено: 27.01.2025 / Одобрено: 05.02.2025 / Опубликовано: 25.12.2025

Цителадзе Д.Д.

Канд. экон. наук, доцент, департамент маркетинга и предпринимательства, ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики», г. Нижний Новгород, e-mail: david.tsiteladze@mail.ru

Шарова А.М.

Младший бизнес-аналитик, ООО «Группа компаний «Иннотех», г. Нижний Новгород

Tsiteladze D.D.

Candidate of Economic Sciences, Associate Professor, Department of Marketing and Entrepreneurship, HSE Nizhny Novgorod, e-mail: david.tsiteladze@mail.ru

Sharova A.M.

Junior Business Analyst, Innotech Group of Companies, Nizhny Novgorod

Аннотация

В работе проведен анализ концепций, методов сбора и трассировки требований в сфере проектного управления, представлена детальная классификация требований, а также проведены сравнения между различными подходами к их сбору и трассировки. Исследование показало, что, несмотря на глубокую проработанность вопросов сбора требований в проектном управлении в научной и прикладной литературе, практически во всех современных рекомендациях и методах сбора требований в проектах присутствуют существенные недостатки, связанные с отсутствием рекомендаций по вопросу формализации задач проекта. В работе показано, что из-за отсутствия подходов, методов и правил формализации задач проекта качество собранных требований оказываются недостаточно высокими. В работе предложен подход, позволяющий решить вопрос формализации задач проекта. Этот подход повышает качество, эффективность и результативность требований к проекту.

Ключевые слова: сбор требований, трассировка требований, проект, матрица трассировки, формализация задач проекта.

Abstract

This paper analyzes concepts, methods, and techniques for collecting and tracing requirements in project management. It presents a detailed classification of requirements and compares various approaches to their collection and tracing. This study revealed that, despite the extensive study of requirements collection in project management in both scientific and applied literature, virtually all current recommendations and methods for collecting project requirements contain significant shortcomings related to the lack of guidance on formalizing project tasks. The paper demonstrates that the lack of approaches, methods, and rules for formalizing project tasks results in insufficient quality of collected requirements and proposes an approach to formalizing project tasks. This approach improves the quality, efficiency, and effectiveness of project requirements.

Keywords: requirements collection, requirements tracing, project, tracing matrix, formalizing project tasks.

Введение

Актуальность данной работы обусловлена существующей на протяжении многих лет проблемой организации процесса работы с требованиями в компаниях, из-за чего некоторые работы не выполняются в срок. Проблема работы с требованиями характерна для рынка России в целом и является одним из основных рисков при реализации проектов. Следовательно, предложение способов решения этой проблемы может позволить снизить количество неудачно реализуемых проектов, а также повысить эффективность использования имеющихся у компании ресурсов.

Процесс работы с требованиями — один из ключевых процессов в проектном управлении, так как его результаты существенно влияют на все дальнейшее течение проекта. На основании собранных

требований команда управления проектами определяет границы и сроки проекта, выстраивает план работ, согласует бюджет проекта, определяет цели проекта и критерии их достижения. Ошибки на этапе сбора требований неизбежно ведут к увеличению стоимости проекта, сроков его выполнения и к снижению качества итогового результата. В конечном итоге эти ошибки могут привести к невозможности достижения стратегических целей компании [1].

Согласно исследованиям, *IBM* около 60% затрат времени команды проектов тратят в результате неэффективной организации процесса управления требованиями [38]. Такие показатели в сложившейся конкурентной ситуации являются неприемлемыми, так как не позволяют эффективно планировать и завершать проекты, эффективно планировать

Таблица 1

ресурсы, находящиеся в дефиците. Получается, что, повысив эффективность этого процесса, можно уменьшить сроки реализации проекта, повысить шансы его успешного завершения и начать делать более точные оценки продолжительности, стоимости и состава работ проекта.

Объектом исследования данной работы являются сложившиеся практики разработки требований и управления ими в компаниях. В качестве предмета исследования выступает процесс работы с требованиями в компании.

Целью данного исследования является систематизация практик работы с требованиями в проектах разработки, внедрения и подготовка на основании этого предложений по совершенствованию процесса работы с требованиями в компаниях.

Задачами данной работы являются:

- 1) определение понятия «Требование» и его место при реализации проекта;
- 2) проведение анализа методов и систематизация методов, применяемых для разработки требований и управления ими в компаниях;
- 3) изучение и анализ процессов работы с требованиями в компании;
- 4) разработка предложений по совершенствованию процесса работы с требованиями в компании.

Анализ подходов и методов сбора требований в проектном управлении

Определение сущности понятия «требование»

Разнообразие проектов и ожиданий от них отразилось на интерпретации термина «требование». В табл. 1 представлены наиболее широко используемые из них.

Анализируя представленные в табл. 1 определения, можно заметить, что они обладают, как общими, так и отличающимися чертами.

Таблица 1

Определение термина «требование» в различных источниках

| Источник | Определение | Комментарий |
|----------|--|--|
| PMBoK | Требования включают в себя условия или возможности, которым должен соответствовать проект или которые должен иметь продукт, услуга или результат, чтобы удовлетворить соглашению или другой формальной предписанной спецификации | Требования включают потребности заказчика и заинтересованных сторон проекта и продукта [1] |

| Источник | Определение | Комментарий |
|--------------------|--|---|
| BABOK | Требование — это пригодное для использования представление потребности | Требования должны быть направлены на понимание того, какую ценность получит заинтересованная сторона после их выполнения. При этом требование может быть выражено как в виде документа, так и в другом, требуемом обстоятельствами виде |
| SWEBOOK | Требование — это свойство, которым должно обладать программное обеспечение, чтобы позволять пользователям решать их проблемы в реальном мире | Требования к программному обеспечению представляют собой комбинацию требований различных людей на разных уровнях организации, которые непосредственно будут использовать программное обеспечение для выполнения своих задач |
| ISO/IEC/IEEE 29148 | Требование — это утверждение, которое выражает потребность и связанные с ней ограничения и условия | Сильное пересечение с определением в BABOK |

В результате можно сделать вывод о том, что многие из приведенных выше определений термина «требование» базируются на потребностях и проблемах заинтересованных лиц, которые необходимо удовлетворить и решить. При этом также в большей части определений под требованиями понимаются непосредственно свойства и функции конечных продуктов проекта.

Если обобщить и дополнить приведенные выше определения, то требование — это исчерпывающее описание представления целей и потребностей заинтересованных лиц проекта и достаточном для их реализации. Это описание того, что должно быть сделано, чтобы удовлетворить потребности заинтересованных лиц, пользователей системы и принести им ожидаемую ценность и пользу.

Рассмотрим в качестве примера составляющие, которые могут включать в себя требования ИТ-проекта:

- 1) описание бизнес-целей заинтересованных лиц проекта;
- 2) описание функций продукта проекта, позволяющих пользователям выполнять поставленные ими задачи;
- 3) описание и прототипы продукта проекта;
- 4) описание ограничений, которые влияют на выбор способов и методов, используемых для реализации продукта проекта.

Кроме перечисленных пунктов в требованиях к продукту проекта может содержаться другая дополнительная информация, которая поможет команде разработчиков лучше понять потребности заинтересованных лиц и пользователей, понять целевой результат.

Важным замечанием является то, что нужно различать между собой требования к проекту и требования к продукту проекта. Требования к проекту обычно являются более обширными и могут содержать данные о бюджете проекта, его сроках, процедурах разработки и выпуска продукта проекта, необходимого для работы команды оборудования и помещений и о многом другом. Требования к проекту не следует включать в документы, в которых описываются требования к продукту проекта. Данная информация не несет ценности для команды проекта, а наоборот, влечет за собой увеличение количества недопониманий, неверных толкований [13].

Подходы к классификации требований

Рассмотрим наиболее часто применяемую классификацию требований к продукту проекта. Согласно ВАВОК можно выделить четыре вида требований [1].

1. Бизнес-требования — это требования, описывающие бизнес-цели заказчика и объясняющие, почему у заказчика возникла потребность в реализации проекта. В требованиях этого уровня отражается бизнес-цель заказчика, но нет ни слова о способах достижения этой цели. Решением бизнес-требования необязательно является разработка, внедрение или доработка продукта проекта, иногда решением может стать изменение текущего бизнес-процесса или проведение комплекса мероприятий, направленных на выполнение поставленной бизнес-цели. Примером бизнес-требования может быть бизнес-цель: увеличить скорость обслуживания заявок клиентов предприятия в 5 раз.
2. Пользовательские требования или требования заинтересованной стороны — это требования, описывающие потребности отдельной заинтересованной стороны или группы стейкхолдеров, которые должны быть удовлетворены разрабатываемым, внедряемым или дорабатываемым программным обеспечением. Пользовательские требования вытекают из бизнес-требований. Из одного бизнес-требования могут следовать несколько пользовательских требований. Примером

пользовательского требования может быть ПО для отдела логистики или производства, которое пользователи должны иметь возможность просматривать в онлайн режиме и видеть все назначенные на них заявки.

3. Требования к решению — это требования, описывающие конкретные характеристики к разрабатываемому, внедряемому или сопровождаемому продукту проекта. Требования к решению делятся на функциональные и нефункциональные требования. Примером функциональных требований может быть необходимость реализации алгоритма действий исполнителей проекта, используя матрицу распределения ответственности RACIS. Примером нефункциональных требований являются требования к надежности, производительности, безопасности, доступности.
4. Переходные требования — это требования, описывающие трансформацию продукта проекта, которыми он должен обладать и которые позволят осуществить переход из текущего состояния в целевое. Переходные требования носят временный характер, они необходимы только на время перехода с одного ПО на другое. Основная цель таких требований: обеспечить успешный переход на новое ПО с учетом остальных видов требований. Примером переходных требований являются правила миграции и конвертации данных, непрерывность деятельности.

В *PMBoK* требования сгруппированы по классам и помимо перечисленных выше четырех классов требований выделяются еще два:

- 1) требования к проекту (ИСО 10 006) — это требования, описывающие условия, которым должен удовлетворять проект. В случае составления требований к продукту проекта данный вид требований следует упустить и не включать в документацию требований к программному обеспечению;
- 2) требования к качеству (ИСО 9001) — это требования СМК, описывающие критерии, которым должен соответствовать конечный результат, чтобы считать его реализацию успешной [6].

В *RUP* требования классифицируются с использованием подхода *FURPS+*, предложенным Робертом Грэйди:

- 1) *functionality* — функциональность;
- 2) *usability* — удобство использования;
- 3) *reliability* — надежность;
- 4) *performance* — производительность;
- 5) *supportability* — поддерживаемость;

б) «+» обозначает возможные ограничения проектирования (*design*), разработки (*implementation*), ограничения на интерфейсы (*interface*) и физические ограничения (*physical*) [7].

Можно заметить, что перечисленные в подходе *FURPS+* и в *SWEBOK* виды требований относятся к требованиям к решению: функциональным и нефункциональным. В данных подходах упускаются требования более высоких уровней.

В *ISO/IEC/IEEE 29148* классификация требований включает в себя три вида требований:

- 1) требования заинтересованных сторон;
- 2) системные требования;
- 3) требования к программному обеспечению [6].

Э. Халл, К. Джексон и Д. Дик предлагают разделять требования на четыре уровня в зависимости от стадии проекта согласно V-модели и тестирования требований на каждой из этих стадий:

- 1) пользовательские требования;
- 2) системные требования;
- 3) требования к подсистемам;
- 4) требования для компонентов.

Анализируя представленные выше классификации требований, можно сделать вывод, что классификация требований по *BABOK* является наиболее общей и всеохватывающей, она обобщает все другие классификации. Остальные классификации либо содержат в себе не все уровни требований, т.е. являются неполными, либо дополняют друг друга и более глубоко детализируют уровни требований, разделяют требования одного уровня по определенным дополнительным признакам. Такая детализация может быть полезной, так как позволит выявить дополнительные требования в процессе анализа.

Этап выявления направлен на идентификацию требований, их сбор, обнаружение и извлечение. Также на данном этапе определяются заинтересованные стороны и непосредственные пользователи продукта проекта, выявляются их цели, задачи. В табл. 2 представлены методы, используемые на этом этапе.

Таблица 2

Методы, часто используемые на этапе выявления требований

| Метод | Описание |
|----------|---|
| Интервью | Разговор, в котором интервьюер задает вопросы заинтересованным лицам, а они отвечают на них. Интервью могут быть организованы как с одним заинтересованным лицом, так и с группой заинтересованных лиц [13] |

| Метод | Описание |
|-------------------------------------|--|
| Семинары | Структурированные совещания, целью которых является выработка заинтересованными лицами совместного решения [4] |
| Фокус-группы | Дискуссия с участием нескольких заинтересованных лиц со схожими интересами, целью которой является определение их идей, мнений, целей и задач [13] |
| Наблюдение | Заключается в наблюдении за действиями пользователей при выполнении их задач. Иногда в процессе наблюдения разрешается задавать уточняющие вопросы, чтобы лучше разобраться в том, почему пользователь поступает так или иначе. Данный метод является очень затратным по времени, поэтому используется на практике не часто [13] |
| Мозговой штурм | Дискуссия, в ходе которой заинтересованные лица высказывают все свои обдуманные и спонтанные идеи по заданной теме. В ходе такой дискуссии может быть обнаружено множество новых идей и, следовательно, требований [1] |
| Опросные листы, Анкетирование | Заключается в предоставлении заинтересованным лицам заранее подготовленного списка вопросов в виде анкеты/опроса [13] |
| Анализ документов | Закключается в чтении и анализе имеющихся регламентов, приказов, инструкций пользователя и других документов [13] |
| Анализ системных интерфейсов | Представляет из себя анализ систем и сервисов, с которыми в ходе своей работы взаимодействует разрабатываемое программное обеспечение [13] |
| Анализ пользовательского интерфейса | Закключается в анализе внешнего вида и поведения существующего программного обеспечения, с которым работают пользователи, заинтересованные лица [1] |
| Анализ рынка | Закключается в сравнении имеющегося процесса и разрабатываемого программного обеспечения с имеющимися на рынке решениями конкурентов [1] |
| Анализ и моделирование процесса | Закключается в изучении существующих процессов, которые планируется автоматизировать или для выполнения которых уже используется какое-либо программное обеспечение, а также в построении схем и моделей этих процессов |
| Построение карты ассоциаций | Представляет из себя схему, построенную заинтересованным лицом, на которой отражены его желания и идеи [13] |
| Прототипирование | Закключается в создании упрощенного прототипа (образца) будущего программного обеспечения с целью подтверждения правильности понимания потребностей и требований заинтересованных сторон [13] |
| Карта стейкхолдеров | Представляет из себя схему, в которой отражены все заинтересованные лица проекта и их интересы. Данный метод направлен на выявление максимального количества заинтересованных лиц проекта, их интересов и степени влияния. Это позволит учесть требования как можно большего количества заинтересованных лиц, не пропустить чьи-либо требования, а также правильно построить работу с заинтересованными лицами |

Окончание табл. 2

| Метод | Описание |
|--|---|
| Совместная разработка приложений (JAD) | Представляет собой семинар, в котором принимают участие основные заинтересованные лица (заказчик проекта, руководитель проекта, эксперты предметной области, ключевые пользователи) и члены команды разработки с целью выработки совместного оптимального решения. Преимущество данного подхода заключается в лучшем понимании конечного результата всеми заинтересованными лицами. Также непосредственно разработчики более глубоко и быстро погружаются в предметную область [13] |

Все перечисленные в табл. 2 методы можно разделить на две группы:

- 1) методы выявления требований, основанные на *непосредственном общении с заинтересованными лицами*;
- 2) *аналитические методы выявления требований*, направленные на анализ текущей ситуации и имеющихся источников.

Эффективность использования тех или иных методов значительно зависит от конкретной ситуации и условий проекта. Например, при сжатых сроках или бюджете использовать метод «Наблюдение» может быть затруднительным, так как на выявление требований таким способом может потребоваться длительное время, а при большом количестве заинтересованных лиц провести интервью с каждым из них может оказаться практически невозможным.

Этап анализа требований заключается в структуризации информации, полученной на этапе выявления. На этом этапе одни требования соотносят с другими, проверяют их соответствие, непротиворечивость, определяют важность и приоритет, детализируют, разделяют по подсистемам и компонентам программного обеспечения.

На этапе анализа применяются методы, направленные на выявление несоответствий, неполноты, противоречивости требований. Например, к таким методам относят:

- 1) построение контекстных диаграмм. Заключается в определении места нового продукта проекта в существующей среде. Помогает определить то, как программное обеспечение будет взаимодействовать с другими устройствами, сервисами, программными обеспечениями, а также то, как пользователи будут взаимодействовать с программным обеспечением [19];
- 2) построение диаграмм бизнес-процессов. Заключается в анализе текущих процессов и в опреде-

лении степени удовлетворения выявленных требований потребностям бизнес-процесса;

- 3) прототипирование. Заключается в предоставлении заинтересованным лицам, пользователям программного обеспечения прототипа с целью выявления несоответствий ожидаемому результату [13];
- 4) анализ осуществимости требований. Заключается в определении целесообразности и возможности реализации требований при имеющихся временных и денежных ресурсах, а также в возможности реализации требований при имеющихся внешних факторах, ограничениях, сложностях технического и архитектурного характера. Здесь также выявляются противоречия между требованиями [13].

Также на этапе анализа осуществляется определение приоритетов требований. Для правильного проставления приоритетов существуют определенные методы. Рассмотрим некоторые из них.

1. *MoSCoW*. Название метода складывается из четырех категорий приоритетности:

- *M — must have*. К этой категории относятся требования с самым высоким приоритетом, которые необходимо выполнить в первую очередь и без реализации которых программное обеспечение либо не будет работать совсем, либо будет работать, но не будет нести пользователям никакой пользы, а значит, не будет им нужно;
- *S — should have*. К этой категории относятся важные требования, которые также являются обязательными для реализации, но не несут такого решающего значения, как требования первой категории;
- *C — could have*. К этой категории относятся желательные требования, которые не являются обязательными для реализации, но могут быть реализованы в случае наличия времени и свободных ресурсов;
- *W — won't have*. К этой категории относятся самые менее приоритетные требования, которые можно либо не реализовывать вообще, либо перенести их реализацию на более поздние периоды. Недостатком метода *MoSCoW* является отсутствие критериев, на основании которых требования можно было бы отнести к той или иной категории приоритетности [13].

2. *Кано-анализ*. Автором данного метода является токийский профессор Нориаки Кано. Данный

метод основан на оценке степени удовлетворенности пользователей, заинтересованных сторон и степени достижения конечного результата от реализации того или иного требования в виде функции программного обеспечения.

В данном методе требования предлагается разделить по приоритетности на четыре категории:

- *threshold characteristics* (пороговые характеристики). К этой категории относятся базовые требования, реализация которых воспринимается как должное. Требования этой категории обязательно должны быть выполнены, так как просто необходимы пользователям, заинтересованным лицам для выполнения их задач. Если не реализовать эти требования, то это может привести к сильной неудовлетворенности пользователей, заинтересованных лиц. В то же время реализация этих требований не несет значительного увеличения удовлетворенности, так как воспринимается как должное. Отсюда следует сложность выявления требований, относящихся к этой категории: о них обычно не говорят, так как воспринимают их наличие как данность;
- *performance characteristics* (эксплуатационные ожидаемые характеристики). К данной категории относятся требования, реализацию которых пользователи и заинтересованные лица ожидают увидеть в программном обеспечении. Эти требования не являются абсолютно необходимыми для функционирования программного обеспечения, но они важны для повышения удовлетворенности пользователей и заинтересованных лиц. Реализация требований, относящихся к этой категории, значительно влияет на степень удовлетворенности. Именно про эти требования обычно говорят пользователи и заинтересованные лица, именно о них они думают в первую очередь;
- *excitement characteristics* (характеристики, вызывающие восхищение). К данной категории относятся требования, описывающие функции программного обеспечения, наличие которых превосходит ожидания пользователей и заинтересованных лиц. О таких требованиях пользователи и заинтересованные лица обычно не говорят и не думают, так как их реализация носит дополнительный, бонусный характер. В то же время реализация таких требований несет значительное увеличение удовлетворенности пользователей и заинтересованных сторон;

- *indifferent characteristics* (индифферентные (безразличные) характеристики). К данной категории относятся требования, реализация которых не увеличивает ценность программного обеспечения для пользователей и заинтересованных лиц, они не хотят их реализации. Реализация таких требований отрицательно влияет на удовлетворенность [1].

На рис. 1 представлен график Кано-анализа. На графике представлена зависимость между удовлетворенностью пользователей и степенью достижения конечного результата, характеризующая каждую категорию требований. При этом на графике отсутствует линия безразличных характеристик, так как они влияют отрицательно и на степень достижения конечного результата, и на удовлетворенность пользователей.



Рис. 1. График Кано-анализа [39]

3. *Четырехуровневая шкала приоритетов*. В данном методе требования делят на четыре категории приоритетности в зависимости от степени их срочности и важности (см. описание «Матрицы Эйзенхауэра»):

- 1) *высокий приоритет*. К данной категории относятся требования, являющиеся и срочными (их необходимо реализовать уже в следующем выпуске), и важными (пользователям и заинтересованным лицам критически необходима функциональность, описываемая требованиями);
- 2) *средний приоритет*. К данной категории относятся требования, являющиеся важными, но не срочными (реализация требований может быть перенесена на следующий выпуск);

- 3) *низкий приоритет*. К данной категории относятся требования, являющиеся не срочными (реализация функциональности может быть перенесена на следующие выпуски, не обязательно, что ближайшие) и не важными (пользователи и заинтересованные лица могут выполнять свои задачи и без реализации функциональности, описанной этими требованиями);
- 4) *реализовывать не требуется*. К данной категории относятся требования, являющиеся, по мнению некоторых заинтересованных лиц, срочными, но не важными. Таким требованиям рекомендуют присваивать самый низкий приоритет или вообще удалять из списка требований.

На рис. 2 представлены четыре категории требований, выделяемых в четырехуровневой шкале приоритетов.

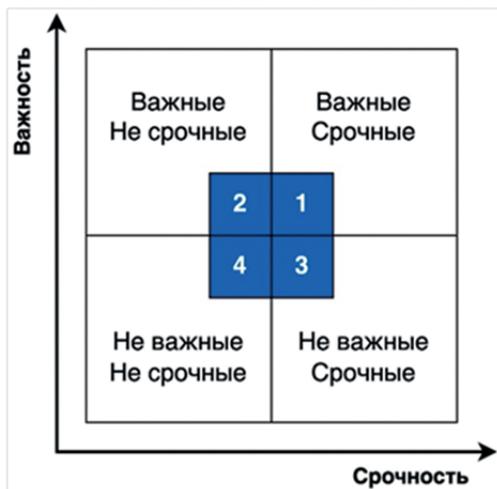


Рис. 2. Категории требований по четырехуровневая шкала приоритетов

Источник: [1].

Этап документирования заключается в преобразовании выявленных и проанализированных требований в письменный вид: в виде текстовых файлов, диаграмм, графиков, схем, таблиц и других способов. В табл. 3 представлены типы документов требований, используемые для документирования разных уровней требований.

Способ документирования требований зависит от типа проекта, принятых в компании стандартов и многих других факторов. Например, в проектах с государственным заказчиком обязательным условием может быть оформление документации по требованиям согласно стандарту ГОСТ. В проектах гибкой разработки требования могут быть задоку-

ментированы в более свободной форме: в виде пользовательских историй (*User Story*), вариантов использования (*Use case*). Также во многих компаниях разрабатываются и применяются собственные шаблоны для документирования требований, включающие только необходимые для компании разделы.

Этап утверждения требований заключается в подтверждении корректности выявленных, проанализированных и задокументированных требований и их готовности для передачи в работу команде разработчиков. Также на этом этапе необходимо разработать критерии приемки, на основании которых созданное программное обеспечение можно будет считать отвечающим требованиям заинтересованных сторон.

Утверждение требований включает в себя их верификацию и валидацию, а также согласование задокументированных требований с заинтересованными лицами.

Верификация требований заключается в проверке задокументированных требований на соответствие отраслевым или принятым в организации стандартам оформления соответствующих документов требований, а также на соответствие требований критериям качества. В процессе верификации нужно ответить на вопрос: «Правильно ли задокументированы требования и оформлены документы требований?»

Таблица 3

Типы документов требований

| Наименование документа | Уровень требований | Описание документа |
|-------------------------------|-----------------------------|--|
| «Концепция и границы» | Бизнес-требования | В документе собираются все выявленные ранее бизнес-требования. В нем фиксируются бизнес-цели, риски, концепция программного обеспечения, его границы, функции, ограничения, контекст, в котором оно реализуется, заинтересованные лица. Образец содержания документа представлен в приложении 1 |
| «Пользовательские требования» | Пользовательские требования | В документе описываются задачи, которые заинтересованное лицо должно решать посредством программного обеспечения. Пользовательские требования могут быть описаны в виде: 1) вариантов использования (<i>use case</i>); 2) диаграммы вариантов использования (<i>use case diagram</i>); 3) пользовательских историй (<i>user story</i>) и т.д. Образцы требований, оформленных в перечисленных выше формах, представлены в приложениях 2–4 |

Окончание табл. 3

| Наименование документа | Уровень требований | Описание документа |
|------------------------------|----------------------|--|
| Спецификация требований к ПО | Требования к решению | В документе описываются функциональные возможности программного обеспечения, его поведение в разных ситуациях и при разных условиях, его свойства, качества, характеристики, а также ограничения на его реализацию |

Валидация требований заключается в проверке задокументированных требований на соответствие целям и потребностям заинтересованных сторон, *бизнес-требованиям*. В процессе валидации нужно ответить на вопрос: «Правильные ли требования были задокументированы?» [1].

Таким образом, верификация отвечает за *внешнюю форму* требований, а валидация — за их *содержание*.

При утверждении требований также рекомендуется использовать критерии качества требований. Существуют критерии качества требований как для *отдельных* требований, так и для *наборов требований*. В разных источниках предлагаются разные критерии качества. В табл. 4 представлены наиболее распространенные критерии качества отдельных требований, а в табл. 5 — критерии качества наборов требований.

На данном этапе также стоит ввести понятие *DoR (definition of ready)*. *DoR* применительно к требованиям представляет из себя критерии готовности требований, которым требования должны удовлетворять, чтобы их можно было передать в работу команде разработки. *DoR* может включать в себя критерии качества требований, но все же является более широким понятием и, помимо критериев качества, может включать в себя дополнительные условия.

Результатом осуществления действий по разработке требований является базовая версия требований (*baseline*) — набор требований, утвержденный и согласованный заинтересованными лицами.

Таблица 4

Критерии качества отдельных требований

| Критерий | Источник | Описание |
|-----------------------------|---|--|
| Атомарность / Независимость | BAVOK, IBM | Требование является независимым от других требований |
| Полнота | BAVOK, Вигерс К. и Битти Д., ISO/IEC/IEEE 29148 | Требование содержит всю необходимую информацию для того, чтобы оно было понято остальными участниками команды и другими заинтересованными лицами и чтобы по нему могла вестись дальнейшая работа |

Окончание табл. 4

| Критерий | Источник | Описание |
|---|--|--|
| Корректность | Вигерс К. и Битти Д., IBM | Требование точно выражает потребности пользователей и заинтересованных лиц и не содержит ошибочной, неверной информации |
| Выполнимость / Осуществимость | BAVOK, Вигерс К. и Битти Д., ISO/IEC/IEEE 29148, IBM | Требование может быть реализовано при имеющихся ресурсах, возможностях и ограничениях |
| Необходимость | Вигерс К. и Битти Д., ISO/IEC/IEEE 29148, IBM | Требование несет ценность и пользу для пользователей и заинтересованных лиц и, действительно, поможет им выполнять свои цели и задачи |
| Приоритизируемость / Назначение приоритетов | BAVOK, Вигерс К. и Битти Д. | Требование уже приоритизировано или может быть приоритизировано с помощью принятого в компании метода приоритизации |
| Однозначность / Недвусмысленность | Вигерс К. и Битти Д., ISO/IEC/IEEE 29148, IBM | Требование может быть интерпретировано только одним способом. Полностью устранить двусмысленность практически невозможно в силу особенностей человеческого языка |
| Тестируемость / Проверяемость | BAVOK, Вигерс К. и Битти Д., ISO/IEC/IEEE 29148, IBM | Реализация требования может быть проверена. Например, на основании требования тестировщик должен суметь составить план тестирования |
| Понятность | BAVOK, IBM | Требование понятно всем заинтересованным лицам и участникам команды, оно написано с использованием терминологии, понятной им |
| Краткость / Ясность | BAVOK, IBM | Требование не содержит лишней, ненужной для его понимания информации |
| Согласованность | BAVOK | Требование описывает действительные потребности пользователей и заинтересованных лиц, а также не противоречит другим требованиям |
| Не содержит описания реализации / Независимость от реализации (Абстрактность) | ISO/IEC/IEEE 29148, IBM | Требование описывает, что должно быть выполнено, а не как. Требование не содержит деталей его реализации |
| Консистентность | ISO/IEC/IEEE 29148 | Требование не противоречит другим требованиям |
| Сингулярность | ISO/IEC/IEEE 29148 | Требование описывает только одну потребность |
| Прослеживаемое (трассируемое) / Элементарность | ISO/IEC/IEEE 29148, IBM | Требование можно проследить от его источника до реализации |

Таблица 5
Критерии качества наборов требований

| Критерий | Источник | Описание |
|---|--|--|
| Полнота / Завершенность | Вигерс К. и Битти Д., ISO/IEC/IEEE 29148, IBM | Набор требований содержит всю необходимую для их реализации информацию и не требует его расширения. В наборе отсутствуют требования, подлежащие определению (TBD), уточнению (TBS), разрешению (TBR) |
| Согласованность / Консистентность / Постоянство | Вигерс К. и Битти Д., ISO/IEC/IEEE 29148, IBM | Требования в наборе требований не противоречат друг другу и не дублируются |
| Способность к модификации | Вигерс К. и Битти Д. | Требования в наборе требований можно изменять без лишних трудностей |
| Отслеживаемость | Вигерс К. и Битти Д., | Требования из набора требований можно проследить от их источника до реализации |
| Доступность | ISO/IEC/IEEE 29148 | Весь набор требований может быть реализован при имеющихся ресурсах, возможностях и ограничениях |
| Ограниченность | ISO/IEC/IEEE 29148 | Набор требований содержит только требования, относящиеся к области планируемого решения, и не выходит за границы этого решения |
| Немногословность | IBM | Требование обозначено один раз, оно не дублирует другие требования и не пересекается с ними |

После получения утвержденного набора требований нужно этими требованиями *управлять*.

Управление версиями требований заключается в идентификации каждой версии уникальным способом с помощью, например, ее наименования определенным образом. Управление версиями включает в себя их отслеживание как отдельных требований, так и их наборов.

Помимо версии, для каждого требования могут быть разработаны и проставлены определенные атрибуты. Например, дата создания требования, его автор, источник, приоритет, состояние и др.

Управление изменениями требований заключается в оценке предлагаемых изменений, их влиянии на ход проекта и принятии решения о внесении этих изменений. Для более эффективной организации процесса управления изменениями каждой компании и команде необходимо разработать и закрепить описание порядка действий при внесении изменений в требования и убедиться, что каждый новый запрос на изменение будет обрабатываться согласно этому описанному порядку. Требования

могут меняться очень часто, и изменять требования по любому запросу любого пользователя нельзя, потому что это влечет за собой ряд проблем, таких как неконтролируемое разрастание границ проекта, неосведомленность всех заинтересованных лиц проекта о внесенных изменениях, увеличение стоимости и сроков проекта и многое другое [13].

Процесс управления изменениями требований начинается либо с обнаружения проблем в существующих требованиях, либо с поступившего запроса на изменение требований [32].

Предлагаются три этапа, из которых должен состоять процесс управления изменениями требований:

- 1) анализ проблем изменения спецификации. Этап заключается в определении обоснованности предложенных изменений;
- 2) анализ изменений и расчет их стоимости. Этап заключается в определении стоимости внесения предложенных изменений с помощью двух показателей: стоимость изменения спецификаций и стоимость изменения непосредственно программного обеспечения. На основании полученных значений показателей определяется целесообразность внесения предложенных изменений;
- 3) реализация изменений. Этап заключается непосредственно во внесении изменений в спецификации и программное обеспечение [32].

Отслеживание состояний требований позволяет определить, на каком этапе реализации находится то или иное требование. При этом состояние требований может входить в атрибутивный состав требований. Список состояний требований и их определения могут отличаться в разных компаниях: некоторым компаниям может потребоваться более широкий список состояний, в то время как другим будет достаточно всего нескольких состояний. Важным замечанием является необходимость фиксации условий для перехода требования из одного состояния в другое, а также строгое соблюдение этих условий: пока не выполнены все условия перехода, требование не должно поменять свое состояние [13].

Отслеживание связей требований позволяет проследить требование от его источника к реализации и наоборот. Можно выделить четыре типа отслеживания требований.

1. *Вперед в направлении к требованиям*. Данный тип отслеживания позволяет определить влияние потребностей пользователей и заинтересованных лиц на требования. Это помогает оценить, какие

из требований могут измениться при изменении тех или иных потребностей пользователей и заинтересованных лиц.

2. *Назад в направлении от требований.* Данный тип отслеживания позволяет определить источник каждого требования: какая потребность пользователей и заинтересованных лиц стала источником того или иного требования.
 3. *Вперед от требований.* Данный тип отслеживания позволяет определить влияние требований на элементы и функции продукта проекта. Это помогает оценить, какие из элементов программного обеспечения могут измениться при изменении тех или иных требований. Также данный тип связи позволяет проверить, что каждое требование было реализовано в программном обеспечении.
 4. *Назад к требованиям.* Данный тип отслеживания позволяет определить причину создания каждого элемента программного обеспечения, определить требование, согласно которому была реализована та или иная функциональность.
- На рис. 3 представлены взаимосвязи перечисленных типов отслеживания.

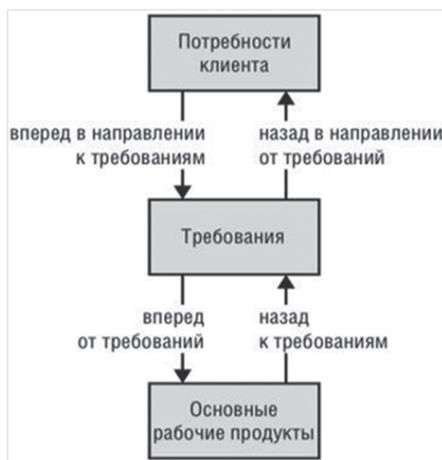


Рис. 3. Взаимосвязи четырех типов отслеживания требований

Источник: составлено автором.

Наиболее известным и часто используемым инструментом для отслеживания требований является матрица отслеживания требований или, по-другому, матрица трассировки требований. Она представляет из себя таблицу, в которой отображаются связи между верхнеуровневыми и нижнеуровневыми требованиями, а также их связь с бизнес-потребностями,

бизнес-целями и конечными элементами, реализованными в программном обеспечении. Такая матрица позволяет определить, несет ли каждое новое требование бизнес-ценность. На рис. 4 представлен пример матрицы отслеживания требований.

| Матрица отслеживания требований | | | | | | | |
|---------------------------------|--------------|---------------------|---|--------------|----------------------------|-----------------------|---------------------|
| Название проекта: | | | | | | | |
| Центр затрат: | | | | | | | |
| Описание проекта: | | | | | | | |
| ID | Связанный ID | Описание требований | Бизнес-цели, потребности, цели и задачи организации | Цели проекта | Поставляемые результаты ИТ | Проектируемый продукт | Контрольные примеры |
| 001 | 1.0 | | | | | | |
| | 1.1 | | | | | | |
| | 1.2 | | | | | | |
| | 1.2.1 | | | | | | |
| 002 | 2.0 | | | | | | |
| | 2.1 | | | | | | |
| | 2.1.1 | | | | | | |
| 003 | 3.0 | | | | | | |
| | 3.1 | | | | | | |
| | 3.2 | | | | | | |
| 004 | 4.0 | | | | | | |
| 005 | 5.0 | | | | | | |

Рис. 4. Пример матрицы отслеживания требований [1]

Алгоритм процесса формализации задач проекта развития

Подходы и методы сбора требований для проектов развития, как правило, представляют собой сложную задачу, которая значительно сложнее, чем для случаев типовых проектов. Подходы и методы сбора требований к проектам, описанные выше, при формальном применении для проектов развития могут оказаться слабо пригодными, а в некоторых случаях абсолютно непригодными. Основная проблема на пути к применению подходов и методов, отмеченных выше, состоит в том, что в них практически отсутствует процедура формализации задач проекта.

Первый и важный шаг, который следует предпринять в ходе процесса формулировки содержания проекта, должен быть связан с критическим анализом этого процесса, который поступил от заказчика проекта. Как правило, формулировка содержания проекта от заказчика требует серьезных уточнений. В этой связи хорошо работает следующая (тайная от заказчика) позиция исполнителя проекта: «заказчик скорее всего сам не знает, чего хочет». После первого шага следует сфокусироваться над вторым шагом — формализацией задачи проекта.

Формализация задач проекта — это процесс представления задач в четкой, понятной и измеримой форме. Этот процесс включает в себя определение цели проекта, анализ альтернативных путей достижения цели проекта, сравнение этих альтернативных

путей, выбор наиболее предпочтительного пути достижения цели на основе стратегических, экономических и тактических ограничений и предпочтений, а также контекста, в котором находится организация в настоящий момент. Однако и этих мероприятий тоже, как правило, недостаточно для того, чтобы посчитать процесс формализации задач проекта завершенным. Обязательно необходимо провести исторический анализ проблемной ситуации, которую необходимо решить в ходе реализации проекта, но и этого тоже недостаточно, чтобы завершить процесс формализации задач проекта. Необходимо провести системный (многофакторный) анализ задач проекта, учитывая внешние и внутренние факторы влияния на фирму в контексте решения задач проекта. В ходе проведения системного анализа необходимо учесть исторический контекст развития факторов влияния, учитывая тренды развития факторов. В завершение этого комплексного процесса (исторического и системного анализа) возможно произвести переход на разбиение проекта на конкретные задачи с описанием их измеримости показателей эффективности, а также определение сроков и ресурсов проекта.

После описания процедуры подготовки к формализации задач проекта укажем методы и инструменты, которые позволят произвести этот процесс эффективно и результативно. Первый и важнейший инструмент — это составление бизнес-кейса проекта (*РМВоК* 6:1.2.6.1). В ходе составления бизнес-кейса проекта необходимо использовать следующие девять ключевых слов (ограничений): соответствие стратегии, содержание, риски, время, деньги, стоимость, качество, выгода, осуществимость. Это позволит применить следующий метод, позволяющий приблизиться к формализации задач проекта — ТОС Э. Голдратта [37]. Применение этого метода позволит определить корневую проблему, решаемую заказчиком проекта. Следующий шаг, который необходимо совершить — это формализация задачи проекта, используя подход, принятый в ТРИЗ Г.С. Альтшуллера и состоящий из четырех пунктов:

- 1) следует сформулировать проблему, которую желает решить заказчик, в ходе реализации проекта, при этом формулировка предметной области проекта должна быть сделана в терминах получения ЦКР — ценностного конечного результата. Здесь важно добиться того, чтобы ЦКР содержал не только инженерную или технологическую ценность, но и обязательно бизнес- и экономиче-

скую ценность проекта. Например, социально значимые проекты, помимо экономической ценности, должны содержать социальную и культурную ценности;

- 2) необходимо сформулировать желаемый конечный результат — ЦКР, который хочет получить заказчик после реализации проекта и ИКР — идеальный конечный результат, к которому следует стремиться в ходе реализации проекта [12];
- 3) необходимо описать МФС — модель функционирующей системы [12]. Здесь необходимо подробно описать систему, в которой наблюдается проблема и в которой следует отметить основные внешние и внутренние факторы, которые влияют на изучаемую систему. В ходе проведения системного анализа можно использовать эффективный ТРИЗ — инструмент, «системный оператор ТРИЗ» или так называемый «девятиэкраннык Г.С. Альтшуллера» [12]. Ниже приведен пример анализа проблемы, связанной с низкой прибыльностью предприятия, где используется «девятиэкраннык»;
- 4) следует отметить ограничения проекта, которые следует учитывать в ходе решения проблемы заказчика (время, стоимость, качество, риски и т.д.).



Рис. 5. Девятиэкраннык Г.С. Альтшуллера для анализа проблемы, связанной с низкой прибыльностью предприятия [12]

На рис. 5 приняты следующие обозначения:

- 1) «система в настоящем» — это задача проекта в настоящем (см. пункт выше);
- 2) «система в будущем» — это желаемый результат проекта (см. пункт 2 выше);
- 3) «надсистема в настоящем» — это внешний фактор влияние на задачу проекта (см. пункт 3 выше);

- 4) «подсистема в настоящем» — это внутренние факторы влияния на задачу проекта (см. пункт 4 выше);
 - 5) оставшиеся обозначения связаны с исторических контекстом развития решения задачи проекта.
- Заключение

В работе систематизированы концепции и методы сбора требований в проектном управлении. Проведённые сравнения между различными подходами сбора и трассировки требований показали их взаимную дополняемость. Показано, что совре-

менные методы сбора требований в проектах имеют основной недостаток, который связан с низким уровнем формализации задач проекта. В работе углублен подход «девятискранника Г.С. Альтшуллера», позволяющий повысить эффективность и результативность сбора требований в проектном управлении посредством более глубокой формализации задач проекта. Этот подход позволяет организовать процесс сбора требований для проектов корректно, продуктивно и в соответствии со стратегией организации.

Литература

1. A Guide to the Business Analysis Body of Knowledge. Version 3. Canada: International Institute of Business Analysis. 2015. 578 с.
2. Agile extension to the BABOK Guide. Version 2. Canada: International Institute of Business Analysis, United States of America: Agile Alliance. 2017. 140 с.
3. Balaji S., Dr. M. Sundararajan Murugaiyan. Waterfall Vs V-model Vs Agile: a comparative study on SDLC // International Journal of Information Technology and Business Management. 29.06.2012. 5 p.
4. Gottesdiener E. Requirements by Collaboration: Workshops for Defining Needs. Addison-Wesley Professional. 2002. 333 с.
5. Guide to the Software Engineering Body of Knowledge. Version 3. IEEE Computer Society. 2016. 335 p.
6. ISO/IEC/IEEE 29148:2011. — Switzerland. 2011. 95 p.
7. Kruchten P. The rational unified process: an introduction. 3rd ed. Addison-Wesley Professional. 2004. 310 p.
8. Mishra A., Dubey D. A comparative study of different Software Development Life Cycle models in different scenarios // International Journal of Advance Research in Computer Science and Management Studies. 5.10.2013. 6 p.
9. PMBoK: руководство к своду знаний по управлению проектами (Руководство PMBoK). — 5-е изд. — Project Management INSTITUTE, Inc., 2013. — 614 с.
10. Zielczynski P. Requirements Management Using IBM Rational RequisitePro. IBM Press/Pearson plc., 2008. 333 p.
11. Вартамян Р.А. Сравнительный анализ методов извлечения требований к программному обеспечению [Текст] / Р.А. Вартамян // Актуальные научные исследования в современном мире. — Переяслав-Хмельницкий [Украина]. — Институт социальной информатики. — 2019. — Вып. 11. — Т. 8. — С. 10–13.
12. Альтшуллер Г.С. Найти идею: Введение в ТРИЗ — теорию решения изобретательских задач [Текст] / Г.С. Альтшуллер. — 11-е изд. — М.: Альпина Паблшер, 2020.
13. Вигерс К., Битти Дж. Разработка требований к программному обеспечению [Текст] / К. Вигерс., Дж. Битти. — 3-е изд., доп.; пер. с англ. — М.: Русская редакция; СПб: БХВ-Петербург, 2014. — 736 с.
14. Вичугова А. Управление требованиями: процессы и стадии жизненного цикла требования к ПО [Текст] / А. Вичугова // Babok-school [сайт] — 2022. — URL: <https://babok-school.ru/blog/requirements-management-processes-and-lifecycle-stages> (дата обращения: 10.02.2024).
15. Горелиц Н.К. Управление требованиями к ответственным системам. Обзор решений [Текст] / Н.К. Горелиц, Д.С. Кильдишев, А.В. Хорошилов // Труды института системного программирования РАН. — 2019. — С. 25–48.
16. ГОСТ 34.201-2020. — М.: Российский институт стандартизации, 2021. — 12 с.
17. ГОСТ 19.701-90. — М.: Стандартинформ, 2010. — 23 с.
18. Гутгарц Р.Д. О формализации функциональных требований в проектах по созданию информационных систем / Р. Д. Гутгарц, Е. И. Провилков. // Киберленка: [сайт] — 2017. — URL: <https://cyberleninka.ru/article/n/o-formalizatsii-funktsionalnyh-trebovaniy-v-proektah-po-sozdaniyu-informatsionnyh-sistem/viewer> (дата обращения: 16.02.2024).
19. Даниярулы Ж. Особенности разработки требований к программному обеспечению [Текст] / Ж. Даниярулы, М. Рыспаева, М. Кажиманова, Ш. Токсынов // Science and technology research, сборник статей II Международной научно-практической конференции. — Петрозаводск: Новая наука, 2021. — С. 47–56.
20. Егоров Ю.С. Выявление и описание пользовательских требований к программному обеспечению [Текст] / Ю.С. Егоров, Д.В. Милов, Е.В. Степанова // Информационные системы и технологии (ИСТ-2017): Материалы докл. XXIII Междунар. науч.-техн. конф. — Нижний Новгород, 2017. — С. 72–76.
21. Итоги 2023 года в российском ИТ / Тинькофф Журнал: [сайт]. — 2023. — URL: <https://journal.tinkoff.ru/news/it-russia-2023> (дата обращения: 20.01.2024).
22. Кравченко Т.К. Управление требованиями при реализации ИТ-проектов [Текст] / Т.К. Кравченко // Бизнес-информатика. — 2013. — С. 63–71.
23. Краснощеков Д.В. Аспекты управления требованиями при разработке программного обеспечения в авиационной отрасли [Текст] / Д.В. Краснощеков, Н.К. Горелиц, Е.В. Песков // ИТ-Стандарт. — 2018. — С. 12–17.
24. Коберн А. Современные методы описания функциональных требований к системам [Текст] / А. Коберн. — М.: Лори, 2002. — 288 с.
25. Леффингуэлл Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход [Текст] / Д. Леффингуэлл, Д. Уидриг; пер. с англ. — М.: Вильямс, 2002. — 448 с.
26. Липко Ю.Ю. К вопросу о методе исследования пользовательских требований к программному обеспечению [Текст] / Ю.Ю. Липко, Д.А. Крымшочалова, З.А. Шогенова, Д.А. Лигидов. // Киберленка: [сайт] — 2021. —

- URL: <https://cyberleninka.ru/article/n/k-voprosu-o-metode-issledovaniya-polzovatelskih-trebovaniy-k-programmnomu-obespecheniyu/viewer> (дата обращения: 15.02.2024).
27. *Маглинец Ю.А.* Анализ требований к автоматизированным информационным системам [Текст]: учеб. пособие / Ю.А. Маглинец. — Интуит НОУ, 2016. — 191 с.
 28. *Паттон Дж.* Пользовательские истории. Искусство гибкой разработки ПО [Текст] / Дж. Паттон. — СПб., 2019. — 364 с.
 29. *Пищикова Е.С.* Техники выявления требований к разработке ПО [Текст] / Е.С. Пищикова, В.Н. Комличенко // Естественные и математические науки в современном мире: сб. ст. XXVI Междунар. науч.-практ. конф. — Новосибирск: СибАК, 2015. — № 26. — С. 42–48.
 30. *Рубцова А.А.* Анализ систем управления требованиями» [Текст] / А. А. Рубцова // Инновационные подходы к решению технико-экономических проблем. — М., 2017. — С. 277–283.
 31. *Рулева Е.Д.* Комплексный подход при разработке требований к программному продукту [Текст] / Е.Д. Рулева // Наука без границ. — 2022. — С. 146–153.
 32. *Соммервилл И.* Инженерия программного обеспечения [Текст] / И. Соммервилл. — 6-е изд. — М.: Вильямс, 2002. — 626 с.
 33. *Спицина И.А.* Системный анализ и моделирование информационных систем [Текст]: учеб. пособие / И.А. Спицина, К.А. Аксенов. — Екатеринбург: Изд-во Уральского университета, 2021. — 118 с. — URL: <https://elar.ufu.ru/handle/10995/100367>
 34. *Титов А.И.* Управление рисками ИТ-проектов на основе компонентной структуры разрабатываемого программного обеспечения [Текст] / А.И. Титов // Киберленка [сайт] — 2017. — URL: <https://cyberleninka.ru/article/n/upravlenie-riskami-it-proektov-na-osnove-komponentnoy-struktury-razrabatyvaемого-programmnogo-obespecheniya/viewer> (дата обращения: 17.02.2024).
 35. *Халл Э.* Разработка и управление требованиями. Практическое руководство пользователя (второе издание) [Текст] / Э. Халл, К. Джексон, Д. Дик. — Танбридж-Уэллс, Кент: Gray Publishing, 2005. — 229 с.
 36. *Шрагенхайм Э.* Теория ограничений в действии. Системный подход к повышению эффективности компании [Текст] / Э. Шрагенхайм. — М.: Альбина Паблишер, 2016. — 286 с.
 37. *Шрагенхайм Э.* Теория ограничений в действии: Системный подход к повышению эффективности компании = Management Dilemmas [Текст] / Э. Шрагенхайм. — М.: Альпина Паблишер, 2014. — 286 с.
 38. What is requirements management? URL: <https://www.ibm.com/think/topics/what-is-requirements-management>
 39. How to use the Kano Model? URL: <https://survalyzer.com/kano-model-analysis>
 4. *Gottesdiener E.* Requirements by Collaboration: Workshops for Defining Needs. Addison-Wesley Professional. 2002. 333 p.
 5. Guide to the Software Engineering Body of Knowledge. Version 3. IEEE Computer Society. 2016. 335 p.
 6. ISO/IEC/IEEE 29148:2011. Switzerland. 2011. 95 p.
 7. *Kruchten P.* The rational unified process: an introduction. 3rd ed. Addison-Wesley Professional. 2004. 310 p.
 8. *Mishra A., Dubey D.* A comparative study of different Software Development Life Cycle models in different scenarios // International Journal of Advance Research in Computer Science and Management Studies. October 5, 2013. 6 p.
 9. PMBoK: A Guide to the Project Management Body of Knowledge (PMBoK Guide). Fifth Edition. Project Management INSTITUTE, Inc. 2013. 614 p.
 10. *Zielczynski P.* Requirements Management Using IBM Rational RequisitePro. IBM Press\Pearson plc. 2008. 333 p.
 11. *Vartanyan R.A.* Comparative Analysis of Software Requirements Extraction Methods // Current scientific research in the modern world. Pereyaslav-Khmelnytsky [Ukraine]. Institute of Social Information. 2019. Issue. 11, vol. 8, p. 12.
 12. *Altshuller G.S.* Find an Idea: Introduction to TRIZ — the Theory of Inventive Problem Solving, 11th ed. Moscow: Alpina Publisher, 2020.
 13. *Wieggers K., Beattie J.* Software Requirements Engineering. 3rd edition, revised. Translated from English. Moscow: Russian Edition Publishing House; St. Petersburg: BHV-Petersburg. 2014. 736 p.
 14. *Vichugova A.* Requirements Management: Processes and Stages of the Software Requirements Life Cycle // Babok-school [website]. 2022. URL: <https://babok-school.ru/blog/requirements-management-processes-and-lifecycle-stages/> (accessed: 10.02.2024).
 15. *Gorelits N.K., Kildishev D.S., Khoroshilov A.V.* Requirements Management for Mission-Critical Systems. Solution Review // Transactions of the Institute for System Programming of the Russian Academy of Sciences. 2019, pp. 25–48.
 16. GOST 34.201-2020. Moscow, Russian Institute of Standardization. 2021. 12 p.
 17. GOST 19.701-90. Moscow, Standartinform. 2010. 23 p.
 18. *Gutgarts R.D., Provilkov E.I.* On the formalization of functional requirements in projects for the creation of information systems // Cyberleninka: [website] 2017. URL: <https://cyberleninka.ru/article/n/o-formalizatsii-funktionalnyh-trebovaniy-v-proektah-po-sozdaniyu-informatsionnyh-sistem/viewer> (accessed: 02/16/2024).
 19. *Daniyaruly Zh., Ryspaeva M., Kazhimanova M., Toksynov Sh.* Features of developing requirements for software // Science and technology research, collected articles of the II International scientific and practical conference. Petrozavodsk: New Science. 2021, pp. 47–56.
 20. *Egorov Yu.S., Milov D.V., Stepanova E.V.* Identification and description of user requirements for software // Information systems and technologies (IST-2017): materials of the report. XXIII Int. scientific and technical. conf. Nizhny Novgorod. 2017, pp. 72–76.
 21. Results of 2023 in Russian IT / Tinkoff Magazine: [website]. 2023. URL: <https://journal.tinkoff.ru/news/it-russia-2023> (accessed: 20.01.2024).
 22. *Kravchenko TK* Requirements management in the implementation of IT projects // Business informatics. 2013, pp. 63–71.
 23. *Krasnoshchekov D.V., Gorelits N.K., Peskov E.V.* Aspects of requirements management in software development in the aviation industry // IT-Standard. 2018, pp. 12–17.

References

1. A Guide to the Business Analysis Body of Knowledge. Version 3. Canada: International Institute of Business Analysis. 2015. 578 p.
2. Agile extension to the BABOK Guide. Version 2. Canada: International Institute of Business Analysis, United States of America: Agile Alliance. 2017. 140 p.
3. *Balaji S., Dr. M. Sundararajan Murugaiyan.* Waterfall Vs V-model Vs Agile: a comparative study on SDLC // International Journal of Information Technology and Business Management. June 29, 2012. 5 p.

24. Koburn A. *Modern Methods for Describing Functional Requirements for Systems*. Moscow: Lori Publishing House. 2002. 288 p.
25. Leffingwell D., Widrig D. *Principles of Working with Software Requirements. A Unified Approach*; translated from English. Moscow: Williams Publishing House. 2002. 448 p.
26. Lipko Yu.Yu., Krymshokalova D.A., Shogenova Z.A., Ligidov D.A. On the method of studying user requirements for software // *Cyberleninka*: [site] 2021. URL: <https://cyberleninka.ru/article/n/k-voprosu-o-metode-issledovaniya-polzovatel'skih-trebovaniy-k-programmnomu-obespecheniyu/viewer> (accessed: 15.02.2024).
27. Maglinets Yu.A. *Analysis of requirements for automated information systems: Tutorial*. Intuit NOU. 2016. 191 p.
28. Patton J. *User Stories. The Art of Agile Software Development*. Saint Petersburg. 2019. 364 p.
29. Pischikova E.S., Komlichenko V.N. *Techniques for Eliciting Requirements for Software Development // Natural and Mathematical Sciences in the Modern World: Coll. Art. XXVI Int. Res. and Pract. Conf. Novosibirsk: SibAK. 2015, no. 26, pp. 42–48.*
30. Rubtsova A.A. *Analysis of Requirements Management Systems // Innovative Approaches to Solving Technical and Economic Problems*. Moscow. 2017, pp. 277–283.
31. Ruleva E.D. *An Integrated Approach to Developing Software Product Requirements // Science without Borders. 2022, pp. 146–153.*
32. Sommerville I. *Software Engineering, 6th edition*. Moscow: Williams Publishing House. 2002. 626 p.
33. Spitsina I.A., Aksyonov K.A. *Systems Analysis and Modeling of Information Systems: A Tutorial*. Ekaterinburg: Ural University Publishing House. 2021. 118 p. URL: <https://elar.urfu.ru/handle/10995/100367>
34. Titov A.I. *Risk Management of IT Projects Based on the Component Structure of Developed Software // Cyberleninka [website] 2017. URL: https://cyberleninka.ru/article/n/upravlenie-riskami-it-proektov-na-osnove-komponentnoy-struktury-razrabatyvaemogo-programmnogo-obespecheniya/viewer* (accessed: 17.02.2024).
35. Hull E., Jackson K., Dick D. *Requirements Engineering and Management. A Practical User's Guide (second edition)*. Tunbridge Wells, Kent: Publisher «Gray Publishing», 2005. 229 p.
36. Schragenheim E. *The Theory of Constraints in Action: A Systems Approach to Improving Company Performance*. Moscow: Albina Publisher, 2016. 286 p.
37. Schragenheim E. *The Theory of Constraints in Action: A Systems Approach to Improving Company Performance = Management Dilemmas*. Moscow: Alpina Publisher, 2014. 286 p.
38. *What is requirements management?* URL: <https://www.ibm.com/think/topics/what-is-requirements-management>
39. *How to use the Kano Model?* URL: <https://survalyzer.com/kano-model-analysis>