

А. А. Кодубец, аспирант, alexey@kodubets.ru, Дальневосточный федеральный университет, Владивосток

Обзор инструментальных средств поддержки в области инженерии требований для программных систем

Представлен краткий обзор инструментальных средств поддержки в области инженерии требований для программных систем. Проведена классификация и дано описание каждого класса инструментальных средств, включая краткое описание соответствующих наиболее функциональных программных решений. Сформулированы достоинства, недостатки и возможные пути улучшения существующих инструментальных средств.

Ключевые слова: инженерия требований, системы управления требованиями, промышленно создаваемые программные системы

Введение

Инженерия требований (*Requirements Engineering*) — это дисциплина, которая занимается систематизацией методов и подходов в области выявления, анализа, разработки, документирования и поддержания требований на протяжении всего жизненного цикла программного обеспечения (ПО). Эта дисциплина является самостоятельным компонентом не только программной инженерии, но и системной инженерии [1, с. 143].

Требования — это утверждения относительно атрибутов, свойств (функций) программной системы, которые необходимы пользователю (заказчику) для решения проблемы или достижения целей [2]. Для описания будущей системы инженерами по требованиям ПО эти утверждения структурируются и документируются в виде текста на естественном языке. Для промышленно создаваемых программных систем процесс разработки требований (*Requirements Development*) является сложным и трудоемким, поскольку такие системы имеют тысячи и более требований [3, с. 179, 4, с. 126, 5, с. 20, 6, с. 5].

Чем сложнее и масштабнее программная система, тем выше риски совершить ошибки в требованиях. Влияние таких ошибок на проект значительное. Например, сравнение исследовательских отчетов группы Стендиша с разницей в 20 лет (1994 и 2014 гг.) [3, с. 3, 7, с. 8] показало, что одной из основных причин успеха проектов было наличие явно задокументированных требований (*Clear Statement of Requirements* — 3-е место среди всех причин, 13 %), а незавершенность требований и отсутствие вовлеченности конечных пользователей стали причиной провала проектов (*Incomplete Requirements, Lack of User involvement*, 1-е и 2-е места среди всех причин, 13,1 и 12,4 % соответственно). Таким образом, требования являются одним из ключевых факторов успеха проектов. В дальнейшем по тексту термины "требования" и "инженерия требований" используются в смысловых значениях требований к программным системам.

С ранних 90-х гг. прошлого века работы в области инженерии требований привели к осознанию необходимости увеличения эффективности разработки требований посредством создания инструментальных средств (*Requirements Management Tools*) для автоматизации рутинных операций, выполняемых инженерами [8]. Как следствие, в 1993 г. вышла первая коммерчески успешная система управления требованиями Rational Doors (сейчас IBM Doors) [8], и затем было создано большое число инструментальных средств в области инженерии требований [9, 11].

Цель и проблема

С момента выхода первых систем управления требованиями сложность программных систем значительно выросла, и вместе с этим возникали новые преграды и проблемы на пути к дальнейшему увеличению эффективности работы инженеров по требованиям ПО. Одна из таких проблем разработки требований для промышленно создаваемых программных систем подробно изложена в работе [12, с. 339—341]: *проблема коммуникации, координации и контроля*. Перечислим далее ключевые факторы, вызывающие данную проблему [12].

1. Для разработки требований необходимо взаимодействие заказчика и исполнителя в целях выяснения деталей и ограничений для создаваемой системы, а также согласование задокументированных требований.

2. Для упрощения разработки программной системы осуществляется архитектурное разбиение целевой системы на подсистемы и компоненты, в том числе проводится иерархическое структурирование работ согласно созданной архитектуре.

3. Структура разделения труда и ответственности может быть достаточно сложной как *вертикально* — заказчик, подрядчики, внутренние или внешние субподрядчики, так и *горизонтально*, когда инфраструктурные компоненты пронизывают различные уровни системы, затрагивая многих участников процесса.

4. С увеличением сложности системы и числа требований увеличивается число представителей заказчика и вовлекаемых внутренних экспертов исполнителя и субподрядчиков.

5. Собирать всех участников процесса вместе по каждому вопросу неэффективно. Собрания и взаимодействие структурируются через специализацию и разделение труда с учетом структуры ответственности и решаемых вопросов.

При таком увеличении числа участников трудозатраты на коммуникацию, координацию и контроль значительно увеличиваются [12, с. 339—341]. Разработка требований начинает зависеть в основном от управления коллективом участников процесса и их эффективного взаимодействия, и таким образом, коммуникация между участниками встает на критический путь процесса разработки требований. Иногда эту проблему называют "*организационные и коммуникационные проблемы инженерии требований*" [13]. Далее для краткости будем использовать сочетание "*проблема коммуникации, координации и контроля*".

Многие инструментальные средства в области инженерии требований *прямо* или *опосредованно* созданы для уменьшения влияния от описанной выше проблемы. Например, инструментарий автоматической проверки правил изложения единичных требований влияет на эффективность и повышает их качество, и, как следствие, более качественные спецификации улучшают коммуникацию между участниками процесса разработки. Однако в данном примере вклад в разрешение проблемы *опосредованный* или *косвенный*, поскольку речь идет именно о качестве требований, а не напрямую об улучшении эффективности *коммуникации*. Другой пример, допустим, спецификация подготовлена для формального рецензирования большим числом экспертов (т. е. координация), и инструментальное средство помогает организовать такое рецензирование с автоматическим напоминанием экспертам о необходимости рецензирования и автору о необходимости исправления замечаний, тем самым упрощая *координацию* и *коммуникацию* между участниками этого процесса. В данном примере вклад в разрешение *проблемы коммуникации, координации и контроля прямой*, поскольку речь идет именно об упрощении координации и коммуникации с экспертами во время рецензирования.

Цель обзора — рассмотрение существующих инструментальных средств поддержки в области инженерии требований для программных систем, а также анализ того, насколько они дают *прямой* вклад в уменьшение или разрешение обозначенной выше проблемы.

Структура обзора

Для поиска инструментальных средств использовались ключевые слова "Система управления требованиями" и "Requirements Management Tool", их аббревиатуры, сокращения и производные в поисковых системах Google и Яндекс, а также поиск по базам данных статей РИНЦ и Scopus. На основе множества источников (в том числе [9—11, 14—19]) был составлен начальный список, содержащий более 150 продуктов. Анализ по-

казал, что даже широко известные инструментальные средства уже заменены на более новые, в том числе, под новыми брендами, поэтому устаревшие продукты убраны из списка. Дополнительно проверено, что поддержка и обновление каждого инструментального средства были не более чем 10 лет назад. После проверок осталось более 80 инструментальных средств в списке. Дальнейшая фильтрация списка осуществлялась по следующим критериям: 1) инструментальное средство должно содержать существенную функциональность, связанную с инженерией требований; 2) имеется поддержка английского либо русского языков; 3) инструментальное средство не имеет значительных дефектов в основных сценариях использования; 4) инструментальное средство предназначено только для гражданского применения. Это позволило сократить список до более 60 поддерживаемых инструментальных средств. Большинство из них коммерческого и иностранного происхождения, а также несколько средств с открытым исходным кодом. На следующем этапе выполнялась разработка классификации через более глубокий анализ информации на сайтах и в документации от поставщиков, а также через дополнительное тестирование (если были доступны пробные версии), и в некоторых случаях через удаленные демонстрации продуктов от поставщиков. Затем для классов с большим числом представителей выполнялась группировка на следующие подклассы: 1) наиболее функциональные продукты, имеющие наибольшее число функций и средств, чем в других продуктах, относительно инженерии требований; 2) другие продукты, предоставляющие только подмножество функций из наиболее функциональных продуктов. На основе выработанной классификации и наиболее функциональных решений каждого класса был подготовлен данный обзор. Итак, выявлены следующие классы:

1) системы управления требованиями (*Requirements Management Tool*);

2) системы управления жизненным циклом продукта и системы управления жизненным циклом программного обеспечения (*Application lifecycle management — ALM* [20] Tool; *Product Lifecycle Management — PLM* Tool; *Software Development Life Cycle — SDLC* Tool);

3) инструментальные средства моделирования и поддержки управления требованиями (*Model-based System Engineering tools* [14]);

4) интеллектуальные системы анализа и повышения качества требований (*Natural Language Processing-based tools for Requirements Analysis*);

5) системы прототипирования пользовательского интерфейса и поддержки управления требованиями (*WYSIWYG-prototyping & Requirements platform*).

В отдельные разделы статьи выделены отечественные инструментальные средства и инструменты с открытым исходным кодом.

Системы управления требованиями

Процесс разработки требований для промышленно создаваемых программных систем включает

в себя ряд типовых активностей, выполняемых инженером по требованиям ПО:

- сбор и выявление ранних требований (пожеланий заказчика), выявление целей и проблем, решаемых задач, определение окружения и ограничений через взаимодействие с представителями заказчика (интервью, опросы, анкетирование), включая импорт пожеланий из документов различных форматов (Excel, Word и т. п.);

- документирование требований, в том числе со структурированием спецификации через использование стандартных шаблонов;

- поиск источников требований, включая анализ предметной области, анализ окружения, аналитика существующих решений, стандартов, государственных регуляций в целях выявления скрытых и производных функциональных и нефункциональных требований;

- идентификация заинтересованных сторон: ответственных лиц, принимающих решение, конечных пользователей и их типов;

- обсуждение пожеланий и требований, согласование их с релевантными внутренними и внешними доменными экспертами, а также экспертами заказчика и координация в случае междисциплинарных вопросов;

- выявление зависимостей от внутренних и внешних субподрядчиков и поставщиков, координация релевантных требований с ними и подготовка компонентных требований для них;

- трассировка системных и компонентных требований, трассировка требований на ранние пожелания заказчика, бизнес-требования, пользовательские требования;

- структурирование работ над требованиями через архитектурное разбиение системы на компоненты, в том числе по необходимости эскизное проектирование и моделирование, и создание предварительной архитектуры в целях выяснения архитектурных и компонентных ограничений, влияющих на требования;

- проверка технической исполнимости требований и верификация характеристик качества единичных требований;

- контроль изменений и версионирование требований;

- задание атрибутов единичных требований (идентификатор, приоритет, источник и т. п.) и приоритизация требований на основе ограничений, в том числе бюджетов и сроков;

- в конечном итоге, валидация и согласование с заказчиком формально задокументированных требований в виде спецификаций.

Данный список может изменяться и дополняться в зависимости от специфики проекта и структуры ответственности. Многие из этих активностей включают в себя рутинные составляющие, которые возможно автоматизировать. Для определения принадлежности к классу "Система управления требованиями" использовался следующий критерий: инструмент должен предназначаться, в первую очередь, для помощи по этим типовым активностям. Итак,

в результате исследования, анализа и тестирования инструментальных средств в области инженерии требований, а также изучения существующих обзоров и рейтингов [9—11, 14—19] были найдены следующие продукты.

Наиболее функциональные:

- Jama Connect [21], выпускается с 2007 г. компанией Jama Software, США;

- Requirements Management Doors [22], выпускается с 1993 г. компанией Rational (сейчас часть IBM), США;

- Dimensions RM [23], выпускается с 2009 г. компанией Serena Software (сейчас часть Micro Focus), США;

- Modern Requirements [24] (ранее inteGREAT), выпускается с 2006 г. компанией Modern Requirements, Канада;

- Visure Requirements, выпускается с 2007 г. компанией Visure Solution [25], США.

Другие системы управления требованиями:

- Accompa Requirements Management Tool [26], выпускается с 2007 г. компанией Accompa, США;

- Requisite RM [27], выпускается с 2015 г. компанией OSSENO Software GmbH, Германия;

- ReqView [28], выпускается с 2013 г. компанией Essam s.r.o., Чехия;

- RequirementOne [29], выпускается компанией RequirementsOne, основанной в 2015 г., Великобритания;

- TraceCloud [30] компании TraceCloud, основанной в 2008 г., США;

- Pearls Pro [31], выпускается с 2017 г. компанией Pearl Inc., Канада.

Наиболее функциональные системы управления требованиями [21—25] помогают по типовым активностям и задачам, упомянутым выше, благодаря наличию следующих средств в них:

- редактора требований с возможностью использования шаблонов, а также задания структуры документа;

- возможности копирования и повторного использования требований;

- импорта пожеланий заказчика из MS Word, MS Excel или IBM Doors, а также экспорта спецификаций требований в MS Word и MS Excel;

- редактора иерархии документов (системные требования, компонентные требования и т. п.);

- редактора атрибутов единичных требований;

- редактора диаграмм для построения моделей (подобно MS Visio);

- версионирования изменений, включая возможность задания публичных и рабочих версий спецификаций;

- инструмента комментирования требований;

- средства построения таблиц трассировки между различными спецификациями требований (бизнес-требования, системные требования, компонентные требования и т. п.), в том числе в целях выяснения влияния изменений на проект (*Impact Analysis*);

- средства подтверждения и валидации требований (соответствия пожеланиям заказчика или бизнес-требованиям);

• инструмента согласования созданных спецификаций требований (Jama [21], Modern Requirements [24]);

• e-mail-нотификации об изменениях;

• наличия поиска и смены представления (View) на основе фильтрации по атрибутам или ключевым словам;

• средства управления рисками;

• редактора жизненного цикла для документов и проектов.

Как видно из списка, наиболее функциональные инструменты имеют значительный объем средств, прежде всего связанных с редактированием и структурированием требований. Эти средства облегчают работу инженеру по требованиям ПО. Однако средства для *прямого* уменьшения или разрешения *проблемы коммуникации, координации и контроля* либо отсутствуют полностью, либо имеют ограниченную функциональность. Например, Jama Connect [21] и Modern Requirements [24] имеют в составе средства комментирования требований, проведения рецензирования, согласования спецификаций и e-mail-нотификации. Однако отсутствуют автоматические средства проведения анкетирования заказчика, его представителей или доменных экспертов, создания фокус-групп для обсуждения подмножеств требований с владельцами (*Requirements owner*) и ответственными командами (*Responsible team*), нет средств ведения структурированной координации, нет средств коммуникации для разрешения выявленных проблем в требованиях, нет средств проверки и контроля зависимостей, и в целом каких-либо средств оперативного контроля. На практике все эти отсутствующие средства компенсируются ручной работой.

Системы управления жизненным циклом продукта и системы управления жизненным циклом программного обеспечения

Разработка требований является одной из стадий жизненного цикла ПО. Если рассматривать жизненный цикл широко, то во время разработки программных систем создается большое число различных сущностей проекта: спецификация требований; архитектура; дизайн компонентов; тесты; программный код; отгружаемые заказчику релизы и т. п. Эти сущности имеют влияние и зависимость друг на друга, поскольку одни создаются на ранних этапах жизненного цикла, другие позднее, но должны полностью согласовываться с предыдущими. Проверка такой согласованности является одним из методов обеспечения качества ПО. Таблицы трассировки между компонентами программной инженерии позволяют выполнить такую проверку. Например, можно протрассировать от теста к требованиям, от дизайна компонента к требованиям и т. п. Есть индустрии, в которых такая трассировка является обязательной, например, автомобильная [32]. Особенно важно использовать таблицы трассировки при изменении требований на поздних фазах, потому что идет каскадное влияние на все сущности проекта.

Для усиления контроля между компонентами программной инженерии в целом на проекте и обеспечения такой трассировки существует класс инструментальных средств, называемый системы управления жизненным циклом ПО (*Software Development Life Cycle — SDLC Tool*) или системы управления жизненным циклом приложений для программных систем (*Application lifecycle management — ALM Tool*) и системы управления жизненным циклом продукта (*Product Lifecycle Management — PLM Tool*) для аппаратных и программно-аппаратных систем. Как правило, ALM- и PLM-продукты являются интеграцией набора самодостаточных модулей, где каждый модуль добавляет поддержку отдельных компонентов программной инженерии или системной инженерии. Для определения принадлежности к данному классу использовался следующий критерий: в инструменте должны быть как минимум средства, связывающие инженерию требований, программирование (его отслеживание) и тестирование. Значимость инструмента определялась по объему именно средств в модуле для инженерии требований. Итак, были найдены следующие программные продукты.

С точки зрения инженерии требований наиболее функциональные ALM-продукты:

• Helix ALM [33] (ранее TestTrack Pro), выпускаемый с 1996 г. компанией Perforce (ранее Seapine Software), США;

• Kovair ALM studio компании Kovair Software [34], основанной в 2008 г., США;

• codeBeamer ALM [35], выпускается с 1999 г. компанией Intland Software GmbH, Германия;

• DevSpec [36], выпускается компанией TechExcel, Inc., основанной в 1995 г., США;

• Valispace [37], выпускается компанией Valispace GmbH, основанной в 2016 г., Португалия.

ALM-продукты для медицинской индустрии:

• Orcanos ALM [38], выпускается компанией Orcanos Ltd., основанной в 2005 г., Израиль;

• Cockpit Enterprise [39], выпускается компаний Cognition Corporation, Великобритания, более 20 лет на рынке;

• Aligned Elements [40], выпускается компанией Aligned AG, основанной в 2006 г., Швейцария;

• MatrixReq ALM [41], выпускается Matrix Requirements GmbH, Германия.

Другие ALM- и PLM-продукты:

• ALM Octane [42], выпускается с 2010 г. (на базе HP Quality Central) компанией Micro Focus (ранее HP), Великобритания;

• Aqua ALM [43], выпускается с 2012 г. компанией Aqua cloud GmbH, Германия;

• ENOVIA PLM [44], выпускается компанией Dassault Systemes (ранее MatrixOne и SmartTeam с 1998 г.), Франция;

• SpiraTeam [45] компании Inflectra, основанной в 2006 г., США;

• Polarion PLM [48] компании Siemens, выпускается с 2015 г., Германия (ранее Polarion Software, США);

• ReqTest [47] выпускается с 2009 г. компанией ReQtest AB, Швеция;

• Rommana — [48], выпускается с 2011 г. компанией Rommana Software, США;

- Sox [49], выпускается компанией EnCo Software GmbH, основанной в 2005 г., Германия;
- Xebrio [50], выпускается компанией Xebrium Software, основанной в 2013 г., США;
- PTC PLM [51], выпускается с 1998 г. компанией PTC, Inc., США.

Наиболее функциональные PLM- и ALM-продукты имеют модуль для разработки и управления требованиями, представляющий собой подмножество средств из решений класса "Системы управления требованиями". Поставщики осознают ограниченность своих ALM- и PLM-продуктов относительно систем управления требованиями, и в качестве компромисса предоставляют средства интеграции с системами управления требованиями [33–35]. Преимущество ALM- и PLM-продуктов прежде всего заключается в интеграции между компонентами программной инженерии и системной инженерии и наличию функциональности шире инженерии требований, перечисленной кратко далее:

- средства управления портфолио проектов и дорожной картой организации;
- средства управления проектом, включая график проекта и его бюджет;
- средства отслеживания дефектов и задач;
- средства управления рисками проекта (*Risk Management*);
- средства управления тестированием (*Test Management*);
- средства трассировки между компонентами программной инженерии (трассировка требований на тесты, требований на дизайн и т. п.);
- функциональность взаимодействия со средствами непрерывной интеграции (*Continuous Integration*), управление релизами и отгрузками;
- интеграция с системами управления версиями и IDE (интегрированная среда разработки — *Integrated Development Environment*);
- интеграция с системами моделирования, импортирование и экспортирование данных из/в форматы других систем;
- средства создания отчетов и просмотра метрик проекта и ПО;
- средства задания модели жизненного цикла ПО и контроля методологии разработки (*Workflow management*);
- средства сопровождения ПО (*Maintenance, Service & Support*).

Более 100 различных интеграций с внешними программными системами имеет Kovaig ALM [34], в том числе выходящие за рамки программной и системной инженерии, в частности, интеграция с ERP (*Enterprise Resource Planning*) и CRM-системами (*Customer Relationship Management*).

В медицинской индустрии есть множество сертификаций для программно-аппаратных систем, например, стандарты ISO 14971 [52] и ISO 13485 [53], в целях отслеживания рисков и требований к безопасности для человека. В связи с этим существует подкласс ALM-продуктов для медицинской индустрии [38–41], имеющие средства (встроенные базы

знаний) для отслеживания соответствия нормативным требованиям.

Программно-аппаратные системы имеют множество ограничений. Например, программные ограничения на используемый размер памяти, время исполнения алгоритмов; аппаратные ограничения по энергопотреблению, размеру и массе изделия; а также экономические ограничения по себестоимости изделия. Эти ограничения могут быть предоставлены заказчиком и прописаны в требованиях, но могут быть и следствием других факторов. Они могут вызываться окружением, спецификой решаемой проблемы, спецификой создаваемого технического решения, или быть связанными с экономическими факторами. Подсистемы и компоненты, в свою очередь, имеют внутренние ограничения. Продукт Valispace [37] позволяет задавать такие ограничения параметров системы и компонентов в виде требований (значений параметров, минимум, максимум) и автоматически вычисляемых формул (подобно PTC MathCAD) на основе параметров дочерних и зависимых компонентов, а сами ограничения автоматически (каскадно) отслеживать как на уровне всей системы в целом, так и на уровне конкретных компонентов. Это важно, поскольку даже небольшие изменения в требованиях могут сильно повлиять на систему в целом. Например, требуется увеличить время работы системы от батареи в 1,5 раза, не меняя емкость батареи и стоимость изделия. Это возможно сделать как за счет программной части, оптимизируя скорость работы алгоритмов, так и за счет аппаратной части, оптимизируя схему питания и т. п., либо комбинируя решения. Таким образом, поддержка в Valispace отслеживания ограничений позволяет повысить эффективность работы инженера по требованиям ПО.

Интеграция различных компонентов программной и системной инженерии в ALM- и PLM-продуктах дает *автоматическую* возможность понять влияние изменений требований на сущности проекта за счет средств трассировки, и как следствие, это уменьшает влияние от *проблемы коммуникации, координации и контроля* на поздних этапах жизненного цикла ПО. Интеграция с внешними системами шире программной инженерии дает возможность *опосредованно* уменьшить проблему, упрощая приоритизацию требований через средства управления бюджетом и графиком проекта. Также, если новая программная система создается как замена существующей системы, которая разрабатывалась с использованием ALM- и PLM-продуктов, то можно использовать аналитику сущностей проекта, т. е. избежать некоторых рисков предыдущего проекта за счет повторного использования знаний и опыта из прошлого проекта. Таким образом, ALM- и PLM-продукты увеличивают эффективность разработки проектов, однако с точки зрения инженерии требований помогают в основном на более поздних этапах жизненного цикла ПО, а на ранних этапах, когда происходит основная работа по разработке требований ПО, эти продукты либо не дают вклада вообще, либо дают низкий вклад в уменьшение основной *проблемы*, описанной в статье.

Инструментальные средства моделирования и поддержки управления требованиями

Требования, моделирование и архитектура системы тесно связаны, особенно на ранних этапах разработки промышленно создаваемой программной системы [54]. Во время построения моделей могут выявляться дополнительные детали и ограничения. На основе требований, цели проекта и решаемой проблемы, а также моделей предметной области, ограничений и других аспектов создается архитектура программной системы. Она влияет на техническую исполнимость требований, а требования влияют на нее [54]. В одной архитектуре возможно выполнить одно подмножество требований, а в другой — другое подмножество. В реализованной и внедренной системе возможны случаи, когда новую функциональность невозможно или слишком затратно реализовать в силу архитектурных ограничений. Моделирование и создание архитектуры являются важными шагами на ранних этапах жизненного цикла.

Существуют инструментальные средства для моделирования и документирования архитектуры, например, MS Visio и IBM Rational Architect [55]. Эти инструменты поддерживают стандартные языки моделирования UML, IDEF и SysML. Инструментальные средства моделирования и поддержки управления требованиями объединяют в себе функциональные возможности систем управления требованиями и инструментальных средств для моделирования, разработки и документирования архитектуры. В англоязычной литературе такие инструменты называют инструментами системного инжиниринга на основе модели-ориентированного подхода или инструментами программной инженерии для разработки, управляемой моделями (*Model-based Systems Engineering tool* [14], *Model-driven development* — MOD [52]). Этот класс инструментов существенно пересекается с UML-инструментами, которых на рынке представлено достаточно много [56]. Поэтому для определения принадлежности инструмента к классу "Инструментальные средства моделирования и поддержки управления требованиями" использовался следующий критерий: в инструменте должны быть как минимум средства, связывающие инженерию требований и моделирование, при этом инструмент должен поддерживать широкий набор стандартов моделирования. Итак, были найдены следующие программные продукты.

Наиболее функциональные:

- Cradle-RM [57], выпускается с 1987 г. компанией 3SL, Великобритания;
- objectiF RM [58], выпускается с 1992 г. компанией microTOOL GmbH, Германия;
- Enterprise Architect [59], выпускается с 2000 г. компанией Sparx Systems Pty Ltd, Австралия;
- Rhapsody [60], выпускается с 1996 г. компанией IBM, США (ранее Rational, США, а еще ранее Telelogic, Швеция и I-Logix Inc, Израиль);
- Cameo System Modeler (ранее MagicDraw) [61], выпускается с 2010 г. компанией Dassault Systemes, Франция (ранее Nomagic, Inc., США).

Другие:

- PREEvision [62], выпускается компанией Vector Informatik GmbH, основанной в 1988 г., Германия;
- MacA&D and WinA&D [63], выпускается компанией Excel Software, основанной в 1986 г., США.

Продукты данного класса имеют значительную функциональность, связанную с моделированием и системным инжинирингом для *программно-аппаратных систем*. Ярким представителем этого класса является Cradle-RM [57]. Он не уступает IBM Rational Architect по функциональности, но при этом имеет дополнительные средства для инженерии требований:

- трассировку от конкретных объектов на моделях (диаграммах) до конкретных требований, тестов или других диаграмм;
- средства представления трассировки в виде диаграмм;
- редактор Requirements Diagram из SysML — т. е. поддержку представления спецификации требований в виде диаграмм;
- поддержку вычисляемых значений в спецификациях, задаваемых формулами, и вывод вычисляемых значений на этапе отображения требований, если заданы зависимые значения (подобно MS Excel);
- средства представления нескольких спецификаций требований в виде единой иерархической диаграммы при наличии связей между ними;
- средства управления рисками — связь рисков и требований;
- средства управления тестированием.

Есть инструментальные средства данного класса, которые специализируются только на автомобильной индустрии и помогают следовать обязательным стандартам и регламентам. Например, решение PREEvision [62], имеющее в одном инструменте функциональность по разработке требований и созданию программных, аппаратных и гибридных программно-аппаратных моделей и архитектур согласно стандарту ISO 26262 [32].

Как было сказано ранее, промышленно создаваемые программные системы имеют большие спецификации требований, множество подсистем и компонентов, которые в свою очередь имеют отдельные спецификации, дизайн и свою архитектуру. Системы данного класса позволяют эффективно связать все эти сущности проекта вместе. Однако данный класс инструментальных средств только *опосредованно* уменьшает влияние от *проблемы коммуникации, координации и контроля* за счет улучшения структурирования информации о создаваемой системе. В большинстве инструментов *прямые* средства уменьшения влияния от *проблемы* либо отсутствуют полностью, либо минимальны и ограничены следующим: возможностью добавлять комментарии к сущностям и e-mail-нотификациям об изменениях. Исключением является Enterprise Architect [59], в котором есть возможность создавать чаты и привязывать их к любым сущностям проектируемой системы, а также проводить рецензирование этих сущностей. Это улучшает сферу коммуникации, но не охватывает все типовые активности, перечисленные ранее, и, как следствие, инженер по требованиям ПО продолжает работать над ними в ручном режиме.

Интеллектуальные системы анализа и повышения качества требований

Разрабатывая требования, инженер анализирует решаемую задачу проекта, проблему и предметную область, все поступающие вводные от заказчика, экспертов для выявления пожеланий и требований и их документирования в виде спецификации на естественном языке, а также согласования их с целями заказчика и исполнителя. Это высокоинтеллектуальная работа. Как показал обзор исследований в области инженерии требований [12, с. 342], существует перспективное направление "Обработка требований как текста на естественном языке" (*Natural Language Processing For Requirements Engineering* [6] — NLP4RE) и соответствующий класс инструментов "Интеллектуальные системы анализа и повышения качества требований", которые решают некоторые задачи инженерии требований с использованием методов машинного обучения, искусственного интеллекта и методов обработки естественного языка (подобно системам проверки правописания). Например, в этих исследованиях и инструментах решаются следующие задачи: проверка изложения единичных требований, поиск дублирования в требованиях, оценка сложности единичных требований и т. п. Итак, были найдены следующие программные продукты данного класса.

Наиболее функциональные:

- QVscribe [64], выпускается компанией QRA Corp, основанной в 2012 г., США;
- Raven for Microsoft Office [65], выпускается компанией Ravenflow, основанная в 2000 г., США;
- VT Docs [65], выпускается компанией Visible Thread, основанной в 2008 г., Ирландия;
- RQA Quality Studio [67], выпускается компанией Knowledge Centric Solutions, S. L., основанной в 1999 г., Испания.

Другие инструменты:

- Requisite QC [68], выпускается с 2014 г. компаний OSSENO Software GmbH, Германия;
- Requirements Quality Assistant [69], является частью IBM Doors Next Generation.

Одним из наиболее функциональных представителей данного класса инструментальных средств является QVscribe. Он дополняет ранее описанные классы инструментальных средств и может быть использован в комбинации с ними: существуют интеграции с MS Word, MS Excel, Jama и др. QVscribe предоставляет следующие *автоматические* средства:

- проверку соответствия изложения требований принятым в индустрии практикам, например, INCOSE [70], EARS [71];
- поиск похожих требований в спецификациях, чтобы избежать дублирования, на основе вычисления дистанции сходства;
- проверку единиц измерения величин, например, чтобы не было указания в одном месте метров, а в другом сантиметров в одном и том же контексте;
- выявление в тексте просторечий, негативного изложения, сложного изложения в предложениях с указанием времени (когда, до, после);

- выявление в тексте степени незавершенности единичных предложений;
- проверку корректности изложения неявных ссылок ("сделать также как там", когда не ясно где это "там");
- проверку в целом степени ясности изложения на основе нахождения неоднозначных слов, выражений и нахождения неясной терминологии;
- возможность предоставления отчетов о качестве требований.

Во время данного исследования автором были привлечены опытные инженеры по требованиям ПО для тестирования QVscribe. В эксперименте QVscribe анализировал несколько десятков страниц текста требований разных проектов. Все автоматически найденные дефекты (46 шт.) были проанализированы инженерами. Результат анализа показал, что QVscribe находит много ошибок первого рода (ложные срабатывания), но так же находит около 30 % настоящих ошибок в изложении. Результаты данного эксперимента показывают, что часть интеллектуальных задач инженера по требованиям ПО может быть автоматизирована.

Другим оригинальным представителем данного класса является Raven for Microsoft Office [65]. Он имеет следующие функциональные средства: автоматический поиск ошибок последовательности изложения и пропусков в требованиях с анализом нарратива (повествования) текста; автоматическое выявление акторов, объектов и зависимостей между ними; автоматическое создание словарей терминов из текста требований; автоматическое создание кросс-функциональных блок-схем из текста требований.

Большинство инструментов данного класса работает с английским языком, и поставщики этих средств позиционируют их на рынке как инструментальные средства для работы со спецификациями требований. Однако RQA Quality Studio [67] может работать со спецификациями, написанными на французском, немецком, японском, испанском, шведском, итальянском и нидерландском языках. А VT Docs [66] позиционируется на рынке шире, чем только анализ спецификаций требований. Например, VT Docs может анализировать запросы на предложения (*Request for Proposal*) и улучшать качество текста контрактов.

Инструментальные средства данного класса помогают справляться с рутинными задачами инженера по требованиям ПО, но дают лишь *опосредованный* вклад в уменьшение *проблемы коммуникации, координации и контроля* за счет улучшения качества изложения требований. Однако методы NLP4RE могут использоваться как компонент или зависимость для разрешения данной *проблемы*. Например, при наличии средств автоматического выявления релевантных экспертов (владельцев) на основе классификации требований методами машинного обучения [72] становится возможным упрощение координации работ, т. е. автоматическое извещение экспертов о требованиях, относящихся к их экспертизе, с запросом обратной связи и т. п.

Системы прототипирования пользовательского интерфейса и поддержки управления требованиями

Промышленно создаваемые системы могут быть без пользовательского интерфейса или с ограниченным пользовательским интерфейсом (UI), например, роботы, автономные системы, где взаимодействие — это конфигурация, которая задается один раз на этапе производства или внедрения. Для такого типа систем нет смысла детально прорабатывать UI. Однако в большинстве своем программные и программно-аппаратные системы имеют интерфейс человеко-машинного взаимодействия. Например, веб-интерфейс, десктоп UI, UI мобильного приложения, дисплей с сенсорным экраном, распознавание жестов и т. п. Сразу несколько вариантов UI из этого списка может поддерживаться в одной системе, включая множество различных экранных форм, которые должны быть реализованы в единообразном стиле на разных платформах, но каждая из платформ может иметь свои ограничения. Разработка эстетической составляющей UI — это ответственность дизайнеров (промышленных дизайнеров, дизайнеров пользовательского интерфейса). Задачи инженера по требованиям ПО, специфичные для области UI, состоят в следующем:

- документирование и анализ пожеланий заказчика относительно UI с проактивным предложением подходов, включая эргономику и учитывая принципы, технологии и знания из инженерной психологии и компьютерной графики;
- оформление концепции и архитектуры UI в виде UML-диаграмм вариантов использования, деятельности и машин состояний и т. п.;
- обсуждение и согласование архитектуры и концепций UI с экспертами и заказчиком;
- прототипирование UI (экранных форм) на основе концепции, уточняя требования, выявляя дополнительные ограничения к создаваемой системе и при необходимости привлекая эстетического дизайнера;
- документирование требований к UI в виде спецификаций, включая экранные формы, либо выделяя их в отдельный документ, и в итоге, согласование этих документов с заказчиком.

Для того чтобы запрототипировать экранные формы существует много инструментальных средств, начиная от средства широкого профиля MS Visio и заканчивая множеством специализированных средств [73] под специфичные платформы или инструменты, совмещенные с IDE, в том числе с поддержкой принципа WYSIWYG. Некоторые из этих инструментов имеют средства поддержки инженерии требований. Для определения принадлежности инструмента к классу "Системы прототипирования пользовательского интерфейса и поддержки управления требованиями" использовался следующий критерий: в инструменте должны быть как минимум средства, связывающие инженерию требований и прототипирование UI. Итак, были найдены следующие программные продукты данного класса:

- Irise [74], выпускается компанией Irise, основанной в 1997 г., США;

- Justinmind [75], выпускается компанией Justinmind, основанная в 2005 г., США.

Irise имеет широкую функциональность для прототипирования UI для веб-приложений, мобильных платформ и десктоп-платформ. Непосредственно для инженерии требований в данном инструменте предусмотрена следующая функциональность:

- редактор требований в контексте WYSIWYG-редактора прототипирования экранных форм с возможностью привязки единичных требований к конкретным элементам на экранной форме;
- интеграция и синхронизация с системами управления требованиями и ALM-системами (Jama, Dimensions RM);
- комментирование требований;
- приоритизация требований;
- поддержка UML и BPMN (*Business Process Model and Notation*);
- контроль версий, включая средства просмотра изменений в требованиях и экранных формах;
- поддержка отправки уведомлений в Skype и Slack;
- импорт требований из MS Excel и экспорт требований в MS Word;
- генератор отчетов по статусу разработки требований (если есть интеграция с ALM-системами);
- средства задания контроля методологии разработки.

Итак, с точки зрения инженерии требований инструменты данного класса имеют малое подмножество функциональности из решений класса "Система управления требованиями", поэтому они и имеют интеграцию с ними. Наличие средств коммуникации — комментирование, веб-интерфейса уведомлений через службы обмена мгновенными сообщениями, e-mail-уведомления — все это улучшает *сферу коммуникации*, но в основе инструментов данного класса находится редактор экранных форм, и в нем отсутствуют механизмы оперативного *контроля и координации* групповой деятельности, которые необходимо выполнять вручную.

Отечественные инструментальные решения

В силу экономических причин иностранные инструментальные средства имеют сравнительно высокую цену для российского рынка. Также ограничением является тот факт, что только некоторые иностранные инструментальные средства [22] локализованы для Российской Федерации (РФ). Исследование показало, что найденные отечественные инструментальные средства для инженерии требований для программных систем имеют небольшую функциональность в сравнении с наиболее функциональными решениями, описанными ранее (Jama [21], Helix ALM [33], CradleRM [57]), но имеют доступную цену [76] и поддержку русского языка и государственных регламентов. Отечественные инструментальные средства рациональнее использовать, когда инженерная документация между заказчиком и исполнителем ведется на русском языке. Итак,

найжены следующие инструментальные средства для программных систем:

Коммерческие:

- Devprom ALM, выпускается с 2008 г. компанией ООО Девпром [76];
- Система управления требованиями на платформе Техэксперт компании АО Кодекс [77], работающей на рынке с 1991 г.

Некоммерческие:

- Requality [78], выпускается с 2010 г. институтом системного программирования им. В. П. Иванникова РАН.

Продукт компании Девпром — это один из наиболее функциональных представителей отечественных решений. Он попадает в класс "Системы управления жизненным циклом программного обеспечения" и имеет средний объем функциональности относительно наиболее функциональных продуктов данного класса. Однако если в этом инструменте переключить язык интерфейса с русского на английский, то существенная часть функциональности становится недоступной.

В рамках анализа отечественных решений ниже представлены инструментальные средства в области инженерии требований для аппаратных систем (механических и электронных изделий):

- Лоцман: PLM [79], выпускается с 2003 г. компанией АСКОН;
- Союз-PLM [80], выпускается с 2010 г. объединением PLM-союз;
- T-Flex RM [81], выпускается с 1992 г. компанией ЗАО Топ Системы.

Отсутствие программно-аппаратных аспектов в этих продуктах является существенным недостатком. Даже в инженерных областях, где традиционно использовались только механические или электронные компоненты, все чаще используются программно-аппаратные решения. Например, автомобильная подсистема для управления впрыском топлива программируется на микроконтроллерах [82]. Программно-аппаратные системы дают гибкость, позволяют создавать системы с алгоритмами нового типа и существенно сокращают жизненный цикл разработки систем [82].

Итак, на текущий момент найденные отечественные инструментальные средства в области инженерии требований для программных систем в основном находятся на начальных этапах развития, но уже могут использоваться для повышения эффективности работы.

Инструментальные средства с открытым исходным кодом

В процессе работы над данным обзором стало понятно, что многие коммерческие продукты (например, Jama Connect [21], IBM Doors [22], Modern Requirements [24], Helix ALM [33] и др.) в области инженерии требований имеют высокую стоимость лицензий на одного пользователя в год, а также — ограничения на минимальное число покупаемых лицензий. Для небольших организаций такая сто-

имость лицензий будет преградой. В связи с этим далее рассмотрены бесплатные инструментальные средства.

В статье [83] рассматривается 20 инструментальных средств в области инженерии требований с открытым исходным кодом и утверждается, что большинство из них либо заброшены авторами, либо имеют низкое качество. В целом это верно, но есть исключения, которые не описаны в работе [83], и будут описаны здесь. Авторы статьи [83] отметили пять проектов. После перепроверки этого только три проекта являются релевантными к ранее описанному классу:

- OSRMT (*Open Source Requirements Management Tool*) [84] — система управления требованиями с Desktop UI, последние изменения в 2020 г., исходный код под лицензией GPL, а с 2014 по 2016 гг. выпускается параллельная версия с Web UI — aNimble Platform [85];
- ReqHeap (*Requirement Heap*) [86] — система управления требованиями, последние изменения в 2013 г., исходный код под лицензией GPL версии 2.0;
- OpenReq!Live [87] — набор инструментов, связанных с инженерией требований, последние изменения в 2021 г., исходный код под лицензией ESL версии 2.0.

Другие инструментальные средства:

- Modelio [88] — инструментальное средство моделирования и поддержки управления требованиями, выпускается с 2009 г. компанией Softeam, Франция, исходный код под лицензией GPL версии 3.0;
- RMF (*Requirements Management for Eclipse*) [89] — система управления требованиями, разрабатывается с 2014 г. некоммерческой организацией Eclipse Foundation, Канада, исходный код под лицензией ESL версии 1.0, последние изменения в 2016 г.;
- ReqT [90] — экспериментальный инструмент, позволяющий задать требования на формальном декларативном языке и автоматически сгенерировать из этого диаграммы (подобно UML/SysML), выпускается с 2010 г. Лундским университетом, Швеция, исходный код под лицензией BSD 2-clause, последние изменения в 2014 г.;
- Fret [91] (*Formal Requirements Elicitation Tool*) — инструмент, позволяющий задать требования на формальном ограниченном языке и автоматически из этого создавать модели для симуляции поведения системы: блок диаграммы или временные диаграммы сигналов (формы волны, Waveform); выпускается с 2019 г. при поддержке NASA, исходный код под лицензией Nasa Open Source Agreement версии 1.3;
- jUCMNav [92] плагин для Eclipse — инструментальное средство моделирования и поддержки управления требованиями, разрабатывается с 2006 г. под лицензией ESL версии 1.0, последние изменения в 2021 г.

Modelio по функциональности имеет значительные средства для моделирования, включая поддержку UML и BPMN. Однако он имеет менее функциональный редактор требований, чем у решений из класса "Системы управления требованиями".

С точки зрения *основной проблемы статьи*, наиболее релевантное инструментальное средство — это OpenReq!Live. Оно разрабатывается группой специалистов четырех университетов различных стран (Финляндия, Испания, Германия, Австрия), а также поддерживается пятью коммерческими компаниями (Siemens, The QT Company и др.). Этот инструмент — это набор экспериментальных модулей (изолированных микросервисов) различных исследователей в области инженерии требований, объединенных под одним брендом. Цель OpenReq — это создание инструментов поддержки разработки требований для открытых систем (*Community based Requirements Engineering*), например, в проектах с открытым исходным кодом, когда нет единого лица принимающего решения. Для этой цели в OpenReq!Live есть следующая экспериментальная функциональность:

- групповое принятие решений через голосование, включая средства определения конфликтов и возможность делегирования своего голоса;
- средства импорта пожеланий из социальных сетей (Twitter) — идея того, что конечные пользователи оставляют обратную связь в социальной сети (жалобы, комментарии) и возможно ее использовать для формирования новых требований;
- средства для автоматического нахождения ответственных лиц на основе данных в системе (истории участия в проектах), использующие NLP4RE методы;
- средства для автоматического нахождения связанных (зависимых) требований, проверки правил изложения единичных требований, нахождения дублирующих требований с использованием NLP4RE-методов.

В большинстве случаев найденные инструментальные средства имеют экспериментальный характер. Они мало полезны на практике в проектах промышленно создаваемых программных систем, за исключением продукта Modelio, который был коммерческим проектом до того, как его исходный код поместили в открытый доступ. Исследовательский проект OpenReq!Live касается *основной проблемы*, описанной в статье. Перечисленные выше средства в нем *прямо* помогают по некоторым аспектам сфер *координации* и *коммуникации*, в частности, голосование и автоматическое нахождение ответственных лиц. Однако в силу ориентации на сообщества разработчиков с открытым исходным кодом данный инструмент не имеет средств для взаимодействия с заказчиком, отслеживания внешних и внутренних зависимостей, отслеживания технической исполнимости требований, и в целом инструмент не имеет средств оперативного контроля. Фактически он является ранним *прототипом* для частичной демонстрации идей, а не готовым продуктом, который можно внедрять.

Другие инструментальные средства

Представленный обзор инструментальных средств сфокусирован на самых больших классах по количеству функциональности. Однако есть и другие инструментальные средства в области инженерии

требований, которые напрямую не попадают в описанные ранее классы. Такие инструменты были рассмотрены, классифицированы и наиболее функциональные представители отмечены далее:

- Agile-инструменты с поддержкой инженерии требований, например, Aha! Roadmap [93], выпускается компанией Aha!, основанной в 2013 г., США;
- инструментальные средства с фокусом на бизнес-моделирование процессов в целях дальнейшего создания информационных систем, например, Blueprint [94], выпускается компанией Blueprint Software Systems, Inc., основанной в 2004 г., Канада;
- плагины для Atlassian JIRA, добавляющие поддержку инженерии требований, например, R4J [95] компании Ease solutions Pte Ltd, основанной в 2007 г., Сингапур;
- плагины для MS Word, добавляющие утилиты для инженерии требований, например, Reqchecker [96] компании Khilogic, Франция;
- редакторы ReqIF-формата (открытый формат для хранения требований Requirements Interchange Format [97]), например, ReqEdit [98] компании ReqTeam GmbH, основанной в 2014 г., Германия;
- средства трассировки документов, в первую очередь требований на дизайн и тесты, например, Reqtify [99], поддерживает более 100 форматов файлов проектных сущностей, выпускается компанией Dassault Systèmes® (до 2013 г. Geensoft), Франция;
- конверторы форматов требований — импорта данных из PDF, MS Word и т. п., и экспорт в ReqIF, IBM Doors и Jama, с возможностью сравнения версий файлов (Diff), например, ReqMan [100] компании em AG, Германия;
- инструменты сбора и оценки идей (*Idea Management*), в конечном итоге, конвертации их в требования, например, Aha! Idea [101], выпускается с 2020 г. компанией Aha!, США.

Заключение

Как отмечено в работе [4], во многих организациях процесс разработки требований выполняется несистемно и неформально — в ручном режиме происходит документирование требований в стандартных офисных пакетах. При увеличении масштаба промышленно создаваемой программной системы дополнительно возникает *проблема коммуникации, координации и контроля*, которая становится на критическом пути процесса разработки требований и только усугубляет ситуацию с ручной работой. Как показал обзор, чтобы помочь с этим существует множество инструментальных средств, в основном представляющих собой специализированный редактор требований и таблицы трассировки для упрощения простых рутинных операций. Большинство инструментов начинали разрабатываться более 10–15 лет назад, когда даже для простых операций было мало автоматизации. Сейчас видно, что покашкоти встречаются с *проблемой коммуникации, координации и контроля*, поскольку в некоторых продуктах добавляются соответствующие средства для упрощения этих аспектов. Для наглядности приве-

дена релевантная функциональность во всех этих инструментальных средствах:

- сфера коммуникации — комментирование единичных требований, комментирование спецификаций, e-mail-нотификации об изменениях, нотификации в мессенджеры, средства импорта пожеланий и жалоб из социальных сетей (экспериментальный прототип [87]);

- сфера координации — рецензирование требований, голосование, автоматическое выявление ответственных лиц (экспериментальный прототип [87]), сбор идей (отдельный продукт [101]);

- сфера контроля — встроенные средства контроля отсутствуют, но у многих инструментов есть интеграция с внешними системами (JIRA), не учитывающие аспекты инженерии требований.

Как следует из списка, существующие инструментальные средства не помогают по всем типовым активностям инженера по требованиям ПО, поскольку в каждом инструменте есть только часть функциональности, приведенной в списке. Поэтому можно сделать вывод, что не существует системного решения проблемы.

Далее перечислены основные недостатки существующих инструментальных средств:

- слабая интеллектуальная составляющая — в большинстве систем не используются достижения методов машинного обучения и искусственного интеллекта, либо используются только для проверки качества единичных требований [64, 66–69], а не для решения *проблемы коммуникации, координации и контроля* (подобно [72]);

- отсутствует или слабо используются базы знаний и связь с предметными областями (за исключением решений для автомобильной [62] и медицинской индустрий [38–41]);

- некоторые из систем [22, 25, 49, 51, 59–65, 67, 75] традиционно разрабатывались как десктоп-приложения и имеют слабый Web UI или не имеют его вообще, аналогично, отсутствует мобильный UI и приложения для мобильных телефонов и планшетов, необходимые для работы в полевых условиях с заказчиком и доменными экспертами, когда нет возможности использовать ноутбук или персональный компьютер;

- ограниченные средства коммуникации — имеются e-mail-нотификации, средства комментирования единичных требований, но отсутствует двусторонняя интеграция с мессенджерами и другими современными средствами связи;

- слабые или отсутствующие механизмы координации с заказчиком и экспертами доменных областей — в некоторых решениях [21, 24, 33, 59] есть средства проведения рецензирования, но нет возможности проведения опросов, создания фокус-групп для решения вопросов технической исполнимости, нет средств создания комитетов для решения междисциплинарных вопросов; многие инструментальные средства ориентированы на одного пользователя, а не на совместную коллективную работу, которая необходима для разработки промышленно создаваемых систем.

- ограниченные средства контроля, за исключением ALM- и PLM-систем, где контроль выполняется через трассировку между компонентами программной и системной инженерии, нет средств оперативного контроля того, что задача выполнена в срок, либо наличие роли контролера в системе с необходимыми средствами поддержки.

Для уменьшения влияния *проблемы коммуникации, координации и контроля* нужны дальнейшие исследования. Необходимо сделать следующее (подобно, как было указано в работе [12, с. 346]): провести анализ и систематизацию структур коммуникации участников процесса, систематизировать типовые структуры координации, выявить активности (коммуникации и координации), дающие наибольший вклад в критический путь процесса разработки требований. На основе этого необходимо создание новых подходов и интегрированного инструментального средства поддержки коммуникации, координации и автоматизированного контроля для инженерии требований, в том числе с исправлением перечисленных выше недостатков существующих инструментальных средств, что сделает возможным сокращение критического пути процесса разработки требований.

Несмотря на все выявленные недостатки существующих инструментальных средств и отсутствие в них системного решения основной *проблемы* статьи, внедрение таких инструментальных средств, в особенности наиболее функциональных, даст экономический эффект и увеличит эффективность организаций, даже с учетом того, что инженер по требованиям ПО продолжит выполнять существенную часть активностей в ручном режиме.

Список литературы

1. Косяков А., Свит У., Сеймур С., Бимер С. Системная инженерия. Принципы и практика / Пер. с англ. под ред. В. К. Батовина. М.: ДМК Пресс, 2014. 636 с.
2. IEEE 610.12—1990 IEEE Standard Glossary of Software Engineering Terminology. URL: https://standards.ieee.org/standard/610_12-1990.html
3. Халл Э., Джексон К., Джерими Д. Разработка и управление требованиями: практическое руководство пользователя, 2-е изд.: Пер. с англ. 2005. 240 с.
4. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. Пер. с англ. М.: Издательский дом "Вильямс", 2002. 448 с.
5. Samer R., Atas M., Felfernig A., Stettinger M., Falkner A., Schenner G. Group Decision Support for Requirements Management Processes // Proceedings of the 20th Configuration Workshop, 2018. P. 19–24.
6. Motger Q., Borrull R., Palomares C., Marco J. OpenReq-DD: A Requirements Dependency Detection Tool // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, 2019. URL: http://ceur-ws.org/Vol-2376/NLP4RE19_paper01.pdf
7. The Standish Group Report — Chaos report // The Standish Group Report — Chaos report. URL: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
8. Ian A. Requirements Management with Doors: A Success Story // Impacts of RE Research on Practice. URL: http://www.scenarioplus.org.uk/papers/doors_story/doors_success_story.htm
9. Requirements Management Tools List — Project Performance International. URL: <https://www.ppi-int.com/wp-content/uploads/2019/04/PPI-005107-8-Requirements-Management-Tools-190403-1.pdf>

10. **Birk A., Heller G.** List of Requirements Management Tools // List of Requirements Management Tools. URL: <https://makingofsoftware.com/resources/list-of-rm-tools/>
11. **Ian A.** Requirements Tools // Ian F Alexander. 2014. URL: <https://scenarioplus.org.uk/vendors.htm>
12. **Кодубец А. А., Артемьева И. Л.** Обзор исследований в области инженерии требований для программных систем // Программная инженерия. 2021. Том 12, № 7. С. 339–349.
13. **Lieble G., Tichy M., Knause E., Ljungkrantz O., Stlegibauer G.** Organization and communication problems in automotive requirements engineering // Requirements Eng 23. 2018. P. 145–167.
14. **MBSE Tools** — Model Based Systems Engineering Wiki. URL: https://www.inco sewiki.info/Model_Based_Systems_Engineering/index.php?title=MBSE_Tools
15. **DPM-10** Best Requirements Management Tools & Software of 2021. URL: <https://thedigitalprojectmanager.com/requirements-management-tools/#overview>
16. **Krotov S.** Top 10 Best Requirements Management (RM) Tools — 2021 | Cllax — Top of IT. URL: <https://www.softwaretestinghelp.com/requirements-management-tools/>
17. **Martin M.** 30 Best Requirements Management Tools in 2021. URL: <https://www.guru99.com/requirement-management-tools.html>
18. **Top 20+** Best Requirements Management Tools (The Complete List). URL: <https://www.softwaretestinghelp.com/requirements-management-tools/>
19. **The 10 Best** Requirements Traceability Tools. URL: <https://www.inflectra.com/tools/requirements-management/10-best-requirements-traceability-tools>
20. **Chappell D.** What is application lifecycle management? Chappell and Associates, Chicago, December, 2008. 6 p.
21. **Product Development Lifecycle Solution | Jama Connect™.** URL: <https://www.jamasoftware.com/platform/jama-connect/>
22. **Engineering Requirements DOORS Family** — Обзор — Российская Федерация | IBM. URL: <https://www.ibm.com/ru-ru/products/requirements-management>
23. **Requirements Management Software | Dimensions RM | Micro Focus.** URL: <https://www.microfocus.com/en-us/products/dimensions-rm/overview>
24. **Requirements Management Tools** — Modern Requirements. URL: <https://www.modernrequirements.com/products/modern-requirements4devops/>
25. **Visure Requirements** — Visure Solutions. URL: <https://visuresolutions.com/requirements-management-tool/>
26. **Accompa** — Requirements Management Software Tool. URL: <https://web.accompa.com/>
27. **ReqSuite® RM** — The Requirements Management Tool that Thinks Ahead. URL: <https://www.osseno.com/en/requirements-management-tool/>
28. **ReqView** — SW & HW Requirements Management Tool / Easy & Flexible. URL: <https://www.reqview.com/>
29. **RequirementONE** Home. URL: <https://www.requirementone.com/index.html>
30. **TraceCloud:** Project Requirements Management & Traceability. URL: <https://www.tracecloud.com>
31. **PEARLS** — The Smartest Requirement Management Tool for BA. URL: <https://pearls-inc.com/>
32. **ISO 26262-1:2018(en)**, Road vehicles — Functional safety — Part 1: Vocabulary. URL: <https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-2:v1:en>
33. **Helix ALM | ALM Tools** | Perforce. URL: <https://www.perforce.com/products/helix-alm>
34. **Application Lifecycle Management** — ALM Tools, Software — Kovair. URL: <https://www.kovair.com/alm-studio/>
35. **Application Lifecycle Management Software** | codeBeamer ALM. URL: <https://intland.com/codebeamer/application-lifecycle-management/>
36. **Requirements Management Software** | Requirements Management Solution. URL: <https://techexcel.com/products/devspec/>
37. **Home** | Valispace. URL: <https://www.valispace.com/>
38. **Orcanos** Application Lifecycle Management Tool & Software — ALM Software Tool — Orcanos Software — ALM And Quality Management. URL: <https://www.orcanos.com/compliance/>
39. **Cockpit Enterprise** | For Medical Device | Cognition Corporation. URL: <https://cognition.us/solutions/cockpit-enterprise/>
40. **Aligned AG** — Aligned Elements — the Medical Device Application Lifecycle Management (ALM) software for Design History File management Corporation. URL: <https://www.aligned.ch/>
41. **Medical Device Design and Documentation Product | Matrix Requirements.** URL: <https://matrixreq.com/>
42. **ALM Octane** — Agile Testing, Release Management & Value Stream Insights | Micro Focus. URL: <https://www.microfocus.com/en-us/products/alm-octane/overview>
43. **Aqua cloud.** URL: <https://aqua-cloud.io/>
44. **ENOVIA 3DEXPERIENCE** — Dassault Systmes®. URL: <https://www.3ds.com/ru/produkty-i-uslugi/enovia/produkty/3dexperience/>
45. **Application Lifecycle Management** — ALM Tools by Inflectra. URL: <https://www.inflectra.com/SpiraTeam/>
46. **Requirements Management, Requirements Gathering, Requirements Management tools** — Polarion REQUIREMENTS. URL: <https://polarion.plm.automation.siemens.com/products/polarion-requirements>
47. **Easiest** Requirements Management Software for IT Teams | ReQtest. URL: <https://reqtest.com/features/requirement-management/>
48. **Rommana** Requirement Management Tools — User Story Tools. URL: <https://rommanasoftware.com/requirement-tools.php>
49. **Software** — EnCo Software GmbH. URL: <https://www.enco-software.com/en/software/>
50. **Requirements Management Software** — Xebrio. URL: <https://www.xebrio.com/requirements-management-software>
51. **Requirements Management and Validation** | PTC. URL: <https://www.ptc.com/en/technologies/plm/requirements-management>
52. **ISO 14971:2019** — Medical devices — Application of risk management to medical devices. URL: <https://www.iso.org/standard/72704.html>
53. **ISO 13485** — Medical devices. URL: <https://www.iso.org/iso-13485-medical-devices.html>
54. **Alebrahim A.** Bridging the Gap between Requirements Engineering and Software Architecture A Problem-Oriented and Quality-Driven Method. Springer Vieweg, 2016. 514 p.
55. **IBM Rational Software Architect Designer** — Overview | IBM ALM. URL: https://www.ibm.com/products/rational-software-architect-designer?mhsrc=ibmsearch_a&mhq=IBM%20Rational%20Architect
56. **25 BEST UML Tools** | FREE UML Diagram Software in 2021. URL: <https://www.guru99.com/best-uml-tools.html>
57. **Cradle** requirements management, alm, mbse, system engineering software, 3SL. URL: <https://www.threesl.com/>
58. **objectiF RM:** Requirements Engineering Software. URL: <https://www.microtool.de/en/products/objectif-rm/>
59. **Full Lifecycle Modeling** for Business, Software and Systems | Sparx Systems. URL: <https://sparxsystems.com/products/ea/index.html>
60. **Engineering Systems Design Rhapsody** — Overview | IBM. URL: <https://www.ibm.com/products/systems-design-rhapsody>
61. **Cameo Systems Modeler** — CATIA — Dassault Systmes®. URL: <https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/>
62. **PREEvision** | The E/E Engineering Solution | Vector. URL: <https://www.vector.com/int/en/products/products-a-z/software/preevision/>
63. **MacA&D and WinA&D** — Requirements Management with Definition Templates, Views & Queries, Reports and Traceability. URL: <https://www.excelsoftware.com/requirementsmanagement>
64. **QVscribe** by QRA Corp — Analyze Requirements Documents in Seconds. URL: <https://qracorp.com/qvscribe/>
65. **RAVEN** for Microsoft Office | Ravenflow. URL: <http://www.ravenflow.com/raven-for-microsoft-office>
66. **VT Docs** — Document Analysis Software for Proposal and Contract Teams. URL: <https://www.visiblethread.com/vt-docs/>
67. **RQA** — QUALITY Studio. URL: <https://www.reusecompany.com/rqa-quality-studio>
68. **ReqSuite® QC** — Automated Quality Control for Requirements. URL: <https://www.osseno.com/en/quality-control/>
69. **Engineering Requirements Quality Assistant** — Overview | IBM. URL: <https://www.ibm.com/products/requirements-quality-assistant>
70. **Guide** for Writing Requirements // Guide for Writing Requirements, INCOSE-TP-2010-006-02.1. URL: <https://connect.incose.org/Pages/Product-Details.aspx?ProductCode=TechGuideWR2019Soft>
71. **Mavin A., Wilkinson P., Harwood A., Novak M.** Easy approach to requirements syntax (EARS) // Requirements Engineering Conference. 2009. P. 317–322.

72. **Samer R., Atas M., Felfernig A., Stettinger M., Falkner A., Schenner G.** Group Decision Support for Requirements Management Processes // Proceedings of the 20th Configuration Workshop, Graz, Austria, September 27th to 28th, 2018. P. 19–24.
73. **25 Best FREE Wireframe Tools & Software in 2021** IBM. URL: <https://www.guru99.com/best-wireframe-tools.html>
74. **Irise** — Best prototyping & requirements platform for remote teams. URL: <https://www.irise.com/>
75. **Free** prototyping tool for web & mobile apps — Justinmind. URL: <https://www.justinmind.com/>
76. **Devprom** ALM — автоматизация процессов разработки ПО. URL: <https://devprom.ru/>
77. **Система** Управления Требованиями (СУТр) — Техэксперт. URL: <https://техэксперт.рф/sutr>
78. **Requality**. URL: <http://www.requality.ru/ru/index.html>
79. **ЛОЦМАН:PLM**. URL: <https://ascon.ru/products/889/review/>
80. **Платформа "Союз-PLM"** | plmsoyuz. URL: <https://www.plmsoyuz.ru/platform>
81. **T-FLEX RM** | Управление требованиями. URL: <https://www.tfex.ru/products/docs/rm/>
82. **Ютт В. Е., Морозов В. В., Чепланов В. И.** Аппараты систем управления зажиганием и впрыском топлива: учеб. пособие. М.: МАДИ, 2013. 112 с.
83. **Santana S., Perero L., Delduca A.** Evaluation of Open Source Tools for Requirements Management // 25th Argentine Congress of Computer Science, CACIC 2019, pp. 188–204.
84. **Open Source Requirements Management Tool** download | SourceForge.net. URL: <https://sourceforge.net/projects/osrmt/>
85. **aNimble** Platform download | SourceForge.net net. URL: <https://sourceforge.net/projects/nimble/>
86. **Requirement** Heap download | SourceForge.net. URL: <https://sourceforge.net/projects/reqheap/>
87. **Requirements Engineering** — tools and solutions offered by OpenReq. URL: <https://openreq.eu/>
88. **Modelio** Open Source — UML and BPMN free modeling tool. URL: <https://www.modelio.org/>
89. **Eclipse** Requirements Modeling Framework. URL: <https://www.eclipse.org/rmf/>
90. **reqT** — a scalable requirements tool. URL: <http://www.reqt.org/>
91. **FRET**: Formal Requirements Elicitation Tool(ARC-18066-1) / NASA Software Catalog. URL: <https://software.nasa.gov/software/ARC-18066-1>
92. **jUCMNav** — WebHome < ProjetSEG < Foswiki — jUCMNav: Juice up your modelling! URL: <http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/WebHome>
93. **Aha!** — The World's #1 Roadmap Software, 2021. URL: <https://www.aha.io/>
94. **Process** Automation Design Platform to Scale RPA | Blueprint. URL: <https://www.blueprintsys.com/>
95. **R4J** — ease solutions. URL: <https://www.easesolutions.com/r4j/>
96. **Reqchecker** — Democratizes requirement engineering. URL: <https://reqchecker.eu/>
97. **About** the Requirements Interchange Format Specification Version 1.2. URL: <https://www.omg.org/spec/ReqIF/>
98. **Products** ReqEdit. URL: <https://www.reqteam.com>
99. **Traceability** | Reqtify — Dassault Systèmes®. URL: <https://www.3ds.com/products-services/catia/products/reqtify/>
100. **ReqMan®** — requests for quotes in record time. URL: <https://www.em.ag/en/reqman/>
101. **Aha!** Software — Idea Managment Software System. URL: <https://www.aha.io/product/ideas>

Requirements Management Tools for Software Systems: a Systematic Tools Review

A. A. Kodubets, e-mail: alexey@kodubets.ru, Far Eastern Federal University, Vladivostok, 690922, Russian Federation

Corresponding author:

Kodubets Alexey A., Postgraduate Student, Far Eastern Federal University, Vladivostok, 690922, Russian Federation
E-mail: alexey@kodubets.ru

Received on October 25, 2021

Accepted on November 18, 2021

This article contains a systematic review of Requirements Management tools (RM tools) for Software Systems. A research question was defined as the following: requirements development process of large-scale software system (with thousands of requirements) and an interaction problem during this process (communication, coordination and control). The problem is caused by the fact that the requirements development process is a cross-disciplinary task and it involves multiple parties — stakeholders, domain experts, and suppliers with own goals and constraints, and thus, the interaction between them seriously slows down the overall requirements development process more than writing the requirements specification itself. The tools were classified into the following classes: Requirements Management Tools, ALM/PLM/SDLC Tools, Model-based System Engineering tools with Requirements support, Natural Language based tools for Requirements Engineering, WYSIWYG-prototyping and Requirements platform, Open Source tools, and tools developed by Russian vendors. Each tool class was described and represented with list of relevant tools — including feature-rich solutions, and a description of key features. A contribution of functionality from each tool into the research question was analyzed and summarized including potential further improvement steps. Advantages and disadvantages of the existing RM tools were represented in this work. To approach the research question, further potential directions were described.

Keywords: requirements engineering, requirements management tools, large-scale software systems

For citation:

Kodubets A. A. Requirements Management Tools for Software Systems: A Systematic Tools Review, *Programmnaya Ingeneria*, 2022, vol. 13, no. 1, pp. 17–31.

DOI: 10.17587/prin.13.17-31

References

1. Kossiakoff A., Sweet W. N., Seymour S. J., Bierner S. M. *System Engineering Principles and Practice 2nd Edition*, Wiley-Interscience, 2011, 531 p.
2. IEEE 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology, available at: https://standards.ieee.org/standard/610_12-1990.html
3. Hull E., Jackson K., Dick J. *Requirements Engineering*, Second Edition, Springer, 2005, 198 p.
4. Leffingwell D., Widrig D. *Managing Software requirements: a use case approach*, Addison-Wesley, Boston, 2003, pp. 17.
5. Samer R., Atas M., Felfernig A., Stettinger M., Falkner A., Schenner G. Group Decision Support for Requirements Management Processes, *Proceedings of the 20th Configuration Workshop*, 2018, pp. 19–24.
6. Motger Q., Borrell R., Palomares C., Marco J. OpenReqDD: A Requirements Dependency Detection Tool, *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium*, 2019. URL: http://ceur-ws.org/Vol-2376/NLP4RE19_paper01.pdf.
7. The Standish Group, Chaos report 2014, available at: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
8. Ian A. Requirements Management with Doors: A Success Story, *Impacts of RE Research on Practice*, available at: http://www.scenarioplus.org.uk/papers/doors_story/doors_success_story.htm
9. Requirements Management Tools List — Project Performance International, available at: <https://www.ppi-int.com/wp-content/uploads/2019/04/PPI-005107-8-Requirements-Management-Tools-190403-1.pdf>
10. Birk A., Heller G. List of Requirements Management Tools, available at: <https://makingofsoftware.com/resources/list-of-rm-tools/>
11. Ian A. Requirements Tools, available at: <https://scenarioplus.org.uk/vendors.htm>
12. Kodubets A. A., Artemieva I. L., Requirements Engineering for Software Systems: A Systematic Literature Review, *Programmya Ingeneria*, 2021, vol. 12, no. 7, pp. 339–349.
13. Lieble G., Tichy M., Knause E., Ljungkrantz O., Stlegibauer G. Organization and communication problems in automotive requirements engineering, *Requirements Eng* 23, 2018, P. 145–167.
14. MBSE Tools — Model Based Systems Engineering Wiki, available at: https://www.incosewiki.info/Model_Based_Systems_Engineering/index.php?title=MBSE_Tools
15. DPM — 10 Best Requirements Management Tools & Software of 2021, available at: <https://thedigitalprojectmanager.com/requirements-management-tools/#overview>
16. Krotov S. Top 10 Best Requirements Management (RM) Tools — 2021 | Cllax — Top of IT, available at: <https://www.softwaretestinghelp.com/requirements-management-tools/>
17. Martin M. 30 Best Requirements Management Tools in 2021, available at: <https://www.guru99.com/requirement-management-tools.html>
18. Top 20+ Best Requirements Management Tools (The Complete List), available at: <https://www.softwaretestinghelp.com/requirements-management-tools/>
19. The 10 Best Requirements Traceability Tools, available at: <https://www.inflectra.com/tools/requirements-management/10-best-requirements-traceability-tools>
20. Chappell D. *What is application lifecycle management?*, Chappell and Associates, 2008, 6 p.
21. Product Development Lifecycle Solution | Jama Connect, available at: <https://www.jamasoftware.com/platform/jama-connect/>
22. Engineering Requirements DOORS Family — Overview — United Kingdom | IBM available at: <https://www.ibm.com/uk/en/products/requirements-management>
23. Dimensions RM — Requirements Management Software Micro Focus, available at: <https://www.microfocus.com/en-us/products/dimensions-rm/overview>
24. Modern Requirements — Requirements Management Tools, available at: <https://www.modernrequirements.com/products/modern-requirements4devops/>
25. Visure Requirements — Visure Solutions, available at: <https://visuresolutions.com/requirements-management-tool/>
26. Accompa — Requirements Management Software Tool, available at: <https://web.accompa.com/>
27. ReqSuite® RM — The Requirements Management Tool that Thinks Ahead, available at: <https://www.osseno.com/en/requirements-management-tool/>
28. ReqView — SW & HW Requirements Management Tool | Easy & Flexible, available at: <https://www.reqview.com/>
29. RequirementONE Home, available at: <https://www.requirementone.com/index.html>
30. TraceCloud: Project Requirements Management & Traceability, available at: <https://www.tracecloud.com>
31. PEARLS — The Smartest Requirement Management Tool for BA, available at: <https://pearls-inc.com/>
32. ISO 26262-1:2018(en), Road vehicles — Functional safety — Part 1: Vocabulary, available at: <https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-2:v1:en>
33. Helix ALM | Perforce, available at: <https://www.perforce.com/products/helix-alm>
34. Kovair ALM Tools — Application Lifecycle Management, Software, available at: <https://www.kovair.com/alm-studio/>
35. codeBeamer ALM — Application Lifecycle Management Software, available at: <https://intland.com/codebeamer/application-lifecycle-management/>
36. Devspec — Requirements Management Software | Requirements Management Solution, available at: <https://techexcel.com/products/devspec/>
37. Valispace, available at: <https://www.valispace.com/>
38. Orcanos Application Lifecycle Management Tool & Software — ALM Software Tool — Orcanos Software — ALM And Quality Management, available at: <https://www.orcanos.com/compliance/>
39. Cockpit Enterprise | For Medical Device | Cognition Corporation, available at: <https://cognition.us/solutions/cockpit-enterprise/>
40. Aligned AG — Aligned Elements — the Medical Device Application Lifecycle Management (ALM) software for Design History File management Corporation, available at: <https://www.aligned.ch/>
41. Matrix Requirements — Medical Device Design and Documentation Product, available at: <https://matrixreq.com/>
42. ALM Octane — Agile Testing, Release Management & Value Stream Insights | Micro Focus, available at: <https://www.microfocus.com/en-us/products/alm-octane/overview>
43. Aqua cloud, available at: <https://aqua-cloud.io/>
44. ENOVIA 3DEXPERIENCE — Dassault Systmes®, available at: <https://www.3ds.com/ru/produkty-i-uslugi/enovia/produkty/3dexperience/>
45. Application Lifecycle Management — ALM Tools by Inflectra, available at: <https://www.inflectra.com/SpiraTeam/>
46. Polarion — Requirements Management, Requirements Gathering, Requirements Management tools, available at: <https://polarion.plm.automation.siemens.com/products/polarion-requirements>
47. ReQtest — Easiest Requirements Management Software for IT Teams, available at: <https://reqtest.com/features/requirement-management/>
48. Rommana Requirement Management Tools — User Story Tools, available at: <https://rommanasoftware.com/requirement-tools.php>
49. Sox — Software — EnCo Software GmbH, available at: <https://www.enco-software.com/en/software/>
50. Xebrio — Requirements Management Software, available at: <https://www.xebrio.com/requirements-management-software>
51. Requirements Management and Validation | PTC, available at: <https://www.ptc.com/en/technologies/plm/requirements-management>
52. ISO — ISO 14971:2019 — Medical devices — Application of risk management to medical devices, available at: <https://www.iso.org/standard/72704.html>
53. ISO — ISO 13485 — Medical devices, available at: <https://www.iso.org/iso-13485-medical-devices.html>
54. Alebrahim A. *Bridging the Gap between Requirements Engineering and Software Architecture A Problem-Oriented and Quality-Driven Method*, Springer Vieweg, 2016, 514 p.
55. IBM Rational Software Architect Designer — Overview, available at: https://www.ibm.com/products/rational-software-architect-designer?mhsrc=ibmsearch_a&mhq=IBM%20Rational%20Architect
56. 25 BEST UML Tools | FREE UML Diagram Software in 2021, available at: <https://www.guru99.com/best-uml-tools.html>
57. Cradle requirements management, alm, mbse, system engineering software, 3SL, available at: <https://www.threesl.com/>
58. objectiF RM: Requirements Engineering Software, available at: <https://www.microtool.de/en/products/objectif-rm/>
59. Enterprise Architect — Full Lifecycle Modeling for Business, Software and Systems | Sparx Systems, available at: <https://sparxsystems.com/products/ea/index.html>

60. **Rhapsody** — Engineering Systems Design Rhapsody — Overview | IBM, available at: <https://www.ibm.com/products/systems-design-rhapsody>
61. **Cameo Systems Modeler** — CATIA — Dassault Systemes®, available at: <https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/>
62. **PREEvision** | The E/E Engineering Solution | Vector, available at: <https://www.vector.com/int/en/products/products-a-z/software/preevision/>
63. **MacA&D** and **WinA&D** — Requirements Management with Definition Templates, Views & Queries, Reports and Traceability, available at: <https://www.excelsoftware.com/requirementsmanagement>
64. **QVscribe** by QRA Corp — Analyze Requirements Documents in Seconds, available at: <https://qracorp.com/qvscribe/>
65. **RAVEN** for Microsoft Office | Ravenflow, available at: <http://www.ravenflow.com/raven-for-microsoft-office>
66. **VT Docs** — Document Analysis Software for Proposal and Contract Teams, available at: <https://www.visiblethread.com/vt-docs/>
67. **RQA** — QUALITY Studio, available at: <https://www.reusecompany.com/rqa-quality-studio>
68. **ReqSuite® QC** — Automated Quality Control for Requirements, available at: <https://www.osseno.com/en/quality-control/>
69. **RQA** — Engineering Requirements Quality Assistant — Overview | IBM, available at: <https://www.ibm.com/products/requirements-quality-assistant>
70. **Ryan M., Wheatcraft L., Zinni R., Dick J., Baksa K.** Guide for Writing Requirements, *INCOSE Publications Office*, 2019.
71. **Mavin A., Wilkinson P., Harwood A., Novak M.** Easy approach to requirements syntax (EARS), *Requirements Engineering Conference*, 2009, pp. 317–322.
72. **Samer R., Atas M., Felfernig A., Stettinger M., Falkner A., Schenner G.** Group Decision Support for Requirements Management Processes, *Proceedings of the 20th Configuration Workshop*, 2018, pp. 19–24.
73. **25 Best FREE Wireframe Tools & Software** in 2021 IBM, available at: <https://www.guru99.com/best-wireframe-tools.html>
74. **Irise** — Best prototyping & requirements platform for remote teams, available at: <https://www.irise.com/>
75. **Justinmind** — Free prototyping tool for web & mobile apps, available at: <https://www.justinmind.com/>
76. **Devprom ALM** — Software Development Process Automation, available at: <https://devprom.ru/>
77. **Requirements Management Tools (SUTr)** — TechExpert, available at: <https://rexэксперт.рф/sutr>
78. **Requality**, Ivannikov Institute for System Programming of the RAS (ISP RAS), available at: <http://www.requality.ru/ru/index.html>
79. **LOCMAN: PLM**, available at: <https://ascon.ru/products/889/review/>
80. **Souz-PLM** platform | plmsoyuz, available at: <https://www.плмсоюз.рф/platform>
81. **T-FLEX RM** | Requirements Management, available at: <https://www.tflex.ru/products/docs/rm/>
82. **Utt V., Morozov V., Cheplanov V.** *Devices of ignition and fuel injection control systems*, Moscow, Madi, 2013, 112 p. (in Russian).
83. **Santana S., Perero L., Delduca A.** Evaluation of Open Source Tools for Requirements Management, *25th Argentine Congress of Computer Science, CACIC 2019*, pp. 188–204.
84. **OSRMT** — Open Source Requirements Management Tool download | SourceForge.net, available at: <https://sourceforge.net/projects/osrmt/>
85. **aNimble** Platform, available at: <https://sourceforge.net/projects/nimble/>
86. **hedHeap** — Requirement Heap, available at: <https://sourceforge.net/projects/reqheap/>
87. **OpenReq R** — Requirements Engineering — tools and solutions offered by OpenReq, available at: <https://openreq.eu/>
88. **Modelio** Open Source — UML and BPMN free modeling tool, available at: <https://www.modelio.org/>
89. **RMF** — Eclipse Requirements Modeling Framework, available at: <https://www.eclipse.org/rmf/>
90. **reqT** — a scalable requirements tool, available at: <http://www.reqt.org/>
91. **FRET: Formal Requirements Elicitation Tool(ARC-18066-1)** | NASA Software Catalog, available at: <https://software.nasa.gov/software/ARC-18066-1>
92. **WebHome** < ProjetSEG < Foswiki — jUCMNav: Juice up your modelling!, available at: <http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/WebHome>
93. **Aha!** — The World's #1 Roadmap Software, available at: <https://www.aha.io/>
94. **Blueprint** — Process Automation Design Platform to Scale RPA, available at: <https://www.blueprintsys.com/>
95. **R4J** — ease solutions, available at: <https://www.easesolutions.com/r4j/>
96. **Reqchecker** — Democratizes requirement engineering, available at: <https://reqchecker.eu/>
97. **ReqIF** — the Requirements Interchange Format Specification Version 1.2, available at: <https://www.omg.org/spec/ReqIF/>
98. **ReqEdit**, available at: <https://www.reqteam.com/en>
99. **Traceability** | Reqtify — Dassault Systemes®, available at: <https://www.3ds.com/products-services/catia/products/reqtify/>
100. **ReqMan®** — requests for quotes in record time, available at: <https://www.em.ag/en/reqman/>
101. **Aha!** Software — Idea Management Software System, available at: <https://www.aha.io/product/ideas>

ИНФОРМАЦИЯ

Продолжается подписка на журнал "Программная инженерия" на первое полугодие 2022 г.

Оформить подписку можно через подписные агентства
или непосредственно в редакции журнала.
Подписной индекс по Объединенному каталогу

"Пресса России" — 22765

Сообщаем, что с 2020 г. возможна подписка
на электронную версию нашего журнала через:

ООО "ИВИС": тел. (495) 777-65-57, 777-65-58; e-mail: sales@ivis.ru,
ООО "УП Урал-Пресс". Для оформления подписки (индекс 013312)
следует обратиться в филиал по месту жительства — <http://ural-press.ru>

Адрес редакции: 107076, Москва, Матросская Тишина, д. 23, оф. 45,
Издательство "Новые технологии",
редакция журнала "Программная инженерия"

Тел.: (499) 270-16-52. E-mail: prin@novtex.ru