

УДК 004.42.05

doi: 10.32620/reks.2020.2.04

А. А. ГОРДЕЕВ

ГВУЗ «Университет банковского дела», Украина

## МОДЕЛЬ КАЧЕСТВА ОТДЕЛЬНОГО ТРЕБОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Основу технического задания на разработку программного обеспечения (ПО) составляет профиль требований, в котором учитываются функциональность, особенности, ограничения, риски и т.д. будущего программного обеспечения. Профиль требований является продуктом процесса профилирования и представляет собой таксономическую структуру, которая связывает между собой множество требований к разрабатываемому программному обеспечению. Неделимой единицей профиля требований является отдельное требование ПО. Формально требование ПО представляет из себя набор связанных предложений, а фактически является более сложным объектом, от качества которого зависит качество ПО в целом. Имплементация в программное обеспечение требования недостаточного качества влечет за собой ресурсные потери. Существующие работы, связанные с этой проблематикой, в полном объеме не предлагают представление модели качества отдельного требования. **Целью статьи** является разработка модели качества отдельного требования ПО. **Объектом исследования** является требование программного обеспечения. Статья посвящена разработке модели качества отдельного требования программного обеспечения. Идея разработки модели базируется на результатах анализа следующих стандартов: ISO/IEC/ IEEE 29148 :2011(E), ISO/IEC/ IEEE 29148 :2018(E) и ISO/IEC 25012:2008. Положения (требования) указанных стандартов учтены при формировании основного материала данной статьи. В ней рассматривается требование, как отдельный несвязанный элемент профиля требований программного обеспечения. Требование представляется в виде элементов фасетно-иерархических структур и состоит из семантического классификационного признака и семантического таксона. Предложена пятикомпонентная модель качества отдельного требования ПО, которая включает структуру, свойства, атрибуты, требования к синтаксису и семантике. Объединение таких элементов в одной модели позволяет формально описать качество отдельного требования ПО. **Как результат**, в данной работе предлагается формальное описание и представление модели качества отдельного требования программного обеспечения.

**Ключевые слова:** требование ПО; профиль требований ПО; качество отдельного требования ПО; фасетно-иерархическая структура; модель качества ПО; свойство требования ПО.

### Введение и постановка задачи

Разработке программного обеспечения должна предшествовать подготовка технического задания и спецификации ПО. Основу таких документов составляет профиль требований, в котором учитываются все особенности, ограничения, риски и т.д. будущего программного обеспечения. Профиль требований является структурой, которая связывает между собой множество требований для разрабатываемого программного обеспечения. Неделимой единицей профиля требований является отдельное требование ПО. Формально требование ПО представляет из себя набор связанных предложений, а фактически является более сложным объектом от качества которого зависит качество ПО в целом. Имплементация в программное обеспечение требования недостаточного качества влечет за собой ресурсные потери.

В существующих работах делались попытки определить проблематику качества требований ПО [1 – 3], рассматривались отдельные случаи качества требований ПО [4], профиля требований [5], а также проводились всесторонние исследования [6 – 8] моделей качества ПО. В существующих работах в полной мере не рассматриваются модели качества отдельного требования ПО.

В связи с этим **целью** статьи является разработка модели качества отдельного требования ПО.

### 1. Формирование модели качества отдельного требования ПО

Под отдельным требованием ПО будем подразумевать само требование и соответствующий ей классификационный признак. Поскольку в статье рассматривается только отдельное требование ПО, в ней не учитываются взаимосвязи между различными требованиями.

Модель качества отдельного требования ПО представлена на рис. 1.

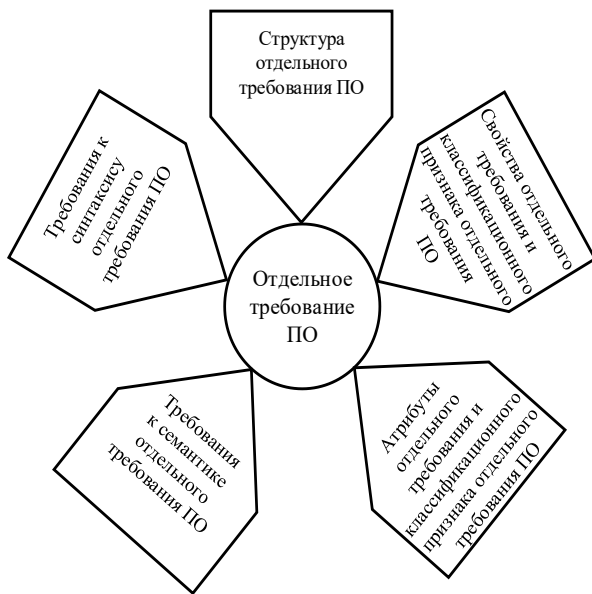


Рис. 1. Составные части модели качества отдельного требования ПО

Данная модель состоит из следующих пяти частей:

- структура отдельного требования ПО;
- свойства отдельного требования ПО и классификационного признака отдельного требования ПО;
- атрибуты отдельного требования ПО и классификационного признака отдельного требования;
- требования к семантике отдельного требования ПО;
- требования к синтаксису отдельного требования ПО.

Рассмотрим более подробно каждую часть модели качества отдельного требования ПО более подробно.

### 1.1. Структура отдельного требования ПО

Отметим, что структура отдельного требования состоит из 2 частей: 1 – требование (семантический таксон), 2 – семантический классификационный признак, соответствующий требованию. В общем виде они являются элементами фасетно-иерархической структуры [9]. Требование ПО и соответствующий ему классификационный признак представляет собой множество связанных между собой слов, которые объединены в отдельные смысловые предложения. Т.е. каждая часть (требование и классификационный признак) состоит из одного и более предложений. Минимальным и неделимым элементом при описании требования и классифика-

ционного признака является слово. В отличие от работ [9-10], в которых рассматривались таксономические структуры без семантического содержания, в данной статье требование является семантическим таксоном, а классификационный признак – семантическим классификационным признаком (рис. 2). В связи с этим при описании элементов требований будем использовать следующие обозначения:

$$1. STE = \langle ste_j \rangle_{j=1}^n - \text{множество элементов семантического таксона (STE – Semantic Taxon Elements) – множество слов в требовании, } ste - \text{слово в требовании, } j - \text{порядковый номер слова в требовании, } n - \text{количество слов в требовании;}$$

2. SCFE =  $\langle scfe_j \rangle_{j=1}^n$  – множество элементов семантического классификационного признака (SCFE – Semantic Classification Feature Elements) – множество слов в классификационном признаке, scfe – слово в классификационном признаке, j – порядковый номер слова в классификационном признаке, n – количество слов в классификационном признаке.

Поскольку порядок слов в предложении является необходимым условием для отображения смысла и его изменение может привести к искажению смысла, элементы (слова) требования и классификационного признака при их формальном описании будут представлены в виде картежей. Таким образом, порядок элементов в представляемых множествах STE (1) и SCFE (2) будет соблюден:

$$STE = \left\langle \begin{matrix} ste_1, ste_2, ste_3, ste_4, ste_5, ste_6, \\ ste_7, ste_8, ste_9, ste_{10}, ste_{11}, ste_{12}, \\ ste_{13}, ste_{14}, ste_{15}, ste_{16}, ste_{17}, ste_{18} \end{matrix} \right\rangle, \quad (1)$$

$$SCFE = \langle scfe_1, scfe_2, scfe_3 \rangle. \quad (2)$$

Стоит отметить, что свойства, атрибуты, требования к семантике и к синтаксису для самого требования ПО и семантического классификационного признака будут различаться. Это связано с отличием в их целевом назначении. Требование более сложная конструкция и задача ее формирования является более ответственной. Задача же семантического классификационного признака точно определять место требования в общей таксономической структуре профиля требований. Соответственно далее элементы модели качества отдельного требования ПО будет рассматриваться отдельно для требования ПО и отдельно для семантического классификационного признака.

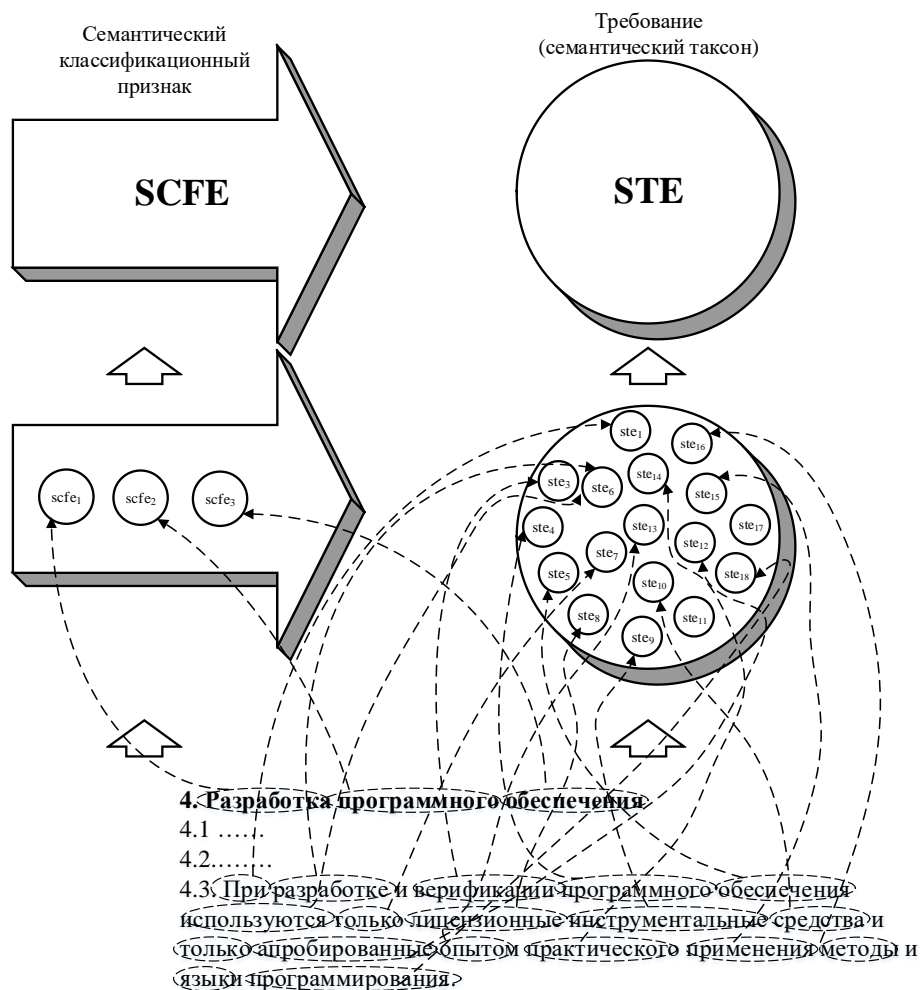


Рис. 2. Структура отдельного требования ПО: от вербального к формальному описанию

## 1.2. Свойства отдельного требования ПО

Сформируем множество свойств для отдельного требования ПО. Для решения такой задачи прежде всего рассмотрим стандарты [11, 12], в которых рассматриваются некоторые свойства качества для отдельного требования. Стандарты [11, 12] предлагают следующий набор свойств: необходимость (necessary), независимость от стоимости реализации (implementation free), однозначность (unambiguous), непротиворечивость (consistent), полнота (complete), исключительность (singular), выполнимость (feasible), трассируемость (traceable), верифицируемость (verifiable), предназначенность (appropriate), корректность (correct), соответствие (conforming).

Предварительный анализ представленных свойств дал следующие результаты:

– 11 из 12 свойств характеризуют отдельное требование ПО. Свойство «непротиворечивость» (consistent) применимо к профилю требований, поскольку основывается на непротиворечивости с другими требованиями в профиле;

– стандарты [8, 9] не содержит полного набора свойств для отдельного требования ПО. Например, в них отсутствуют такие свойства как «актуальность» (currentness), «понятность» (understandability) и др.

Для формирования более полной номенклатуры свойств качества отдельного требования рассмотрим дополнительный стандарт [13], в котором описывается модель качества данных. Соотнесем свойства качества для отдельных требований из [11, 12] со свойствами качества, данных из [13], и результат такого соотнесения представим в табл. 1. Полученные результаты позволили получить общее более полное множество свойств качества для отдельного требования ПО. В табл. 1 свойства для отдельного требования ПО пронумерованы по порядку (крайний правый столбец), причем если в строке со свойством вместо порядкового номера содержится пустая ячейка, считается, что такое свойство не вошло в общее множество свойств качества отдельного требования ПО. Полученное множество свойств из обозначенных стандартов было сформировано по следующим принципам:

Таблица 1  
Свойства модели качества требования ПО

№ п/п	Название характеристики	[10]	[9]	№ п/п
1	Точность (Accuracy)	+		1
2	Полнота (Completeness)	+	+	2
3	Последовательность (Consistency)	+		3
4	Достоверность (Credibility)	+		4
5	Актуальность (Currentness)	+		5
6	Доступность (Accessibility)	+		6
7	Согласованность (Compliance)	+		7
8	Конфиденциальность (Confidentiality)	+		8
9	Эффективность (Efficiency)	+		
10	Прецизионность (Precision)	+		
11	Отслеживаемость (Traceability)	+	+	9
12	Понятность (Understandability)	+		10
13	Готовность (Availability)	+		
14	Портативность (Portability)	+		
15	Восстанавливаемость (Recoverability)	+		
16	Необходимость (Necessary)		+	11
17	Независимость от стоимости реализации (Implementation Free)		+	12
18	Однозначность (Unambiguous)		+	13
19	Непротиворечивость (Consistent)		+	
20	Исключительность (Singular)		+	14
21	Выполнимость (Feasible)		+	15
22	Верифицируемость (Verifiable)		+	16
23	Предназначенность (Appropriate)		+	17
24	Корректность (Correct)		+	18
25	Соответствие (Conforming)		+	19

– принадлежность свойств из [11, 12] к отдельному требованию ПО. В связи с этим в общую номенклатуру свойств вошли только 11 из 12 свойств из стандартов [11, 12] за исключением свойства «непротиворечивость» (consistent), которое свойственно только профилю требований. Таким образом, из [11, 12] вошли следующие свойства «необходимость» (necessary), «независимость от стоимости реализации» (implementation free), «однозначность» (unambiguous), «полнота» (complete), «исключительность» (singular), «выполнимость» (feasible), «верифицируемость» (verifiable), «предназначенность» (appropriate), «корректность» (correct), «соответствие» (conforming);

– соответствие свойств модели качества данных [10] модели качества отдельного требования ПО. В связи с таким принципом 10 из 15 свойств [13] могут характеризовать отдельное требование ПО, а именно «точность» (accuracy), «полнота» (completeness), «последовательность» (consistency), «достоверность» (credibility), «актуальность» (currentness), «доступность» (accessibility), «согласованность» (compliance), «конфиденциальность» (confidentiality), «отслеживаемость» (traceability), «понятность» (understandability).

Стоит также отметить, что два свойства «полнота» (completeness) и «отслеживаемость» (Traceability), которые вошли в общую номенклатуру свойств для отдельного требования ПО присутствуют в двух стандартах.

Введем формальные обозначения для таких свойств:  $SIRC = \{sirc_j\}_{j=1}^{19}$  – множество свойств отдельного требования программного обеспечения (SIRC – Software Individual Requirement Characteristics),  $sirc$  – свойство отдельного требования программного обеспечения,  $j$  – порядковый номер свойства отдельного требования программного обеспечения.

Определим множество свойств для классификационного признака отдельного требования ПО. Для этого рассмотрим следующие свойства:

1. Уникальность (uniqueness) – делает объект классификации отличным от группы однородных объектов. Чем более уникален классификационный признак, тем более неопровержимо он подтверждает тождество объекта;

2. Выраженность (evidence) – позволяет обнаружить и зафиксировать классификационный признак;

3. Устойчивость (воспроизводимость) (stability) – дает возможность зафиксировать классификационный признак спустя время в первоначальном виде, т.е. он должен иметь способность к многократному отображению;

4. Простота обнаружения (simplicity of discovery) – сравнительная простота обнаружения классификационного признака среди всего множества классификационных признаков;

5. Позиционность (positioning) – занимает соответствующее только ему место среди всего множества классификационных признаков;

6. Точность (accuracy) – точно передает значение (смысл) классификационного признака;

7. Согласованность (compliance) – обеспечивает необходимую согласованность между объектами классификации и классификационным признаком;

8. Понятность (understandability) – возможность читать и интерпретировать значение классификационного признака, представленного в виде соответствующих символов и языковых единиц в конкретном контексте использования;

9. Однозначность (univocacy) – обеспечивает единственную однозначную интерпретацию значения классификационного признака;

10. Непротиворечивость (consistency) – не противоречит другим классификационным признакам.

Предварительный анализ потенциальных свойств классификационного признака позволил определить множество свойств, которые свойственны только классификационному признаку отдельного требования ПО. К таким свойствам относятся: 1. «Выраженность» (evidence), 2. «Устойчивость» (stability), 3. «Точность» (accuracy), 4. «Согласованность» (compliance), 5. «Понятность»

(understandability), б. «Однозначность» (univocacy). Таким образом каждый классификационный признак для отдельного требования ПО должен соответствовать 6 свойствам. Введем формальные обозначения для таких свойств:  $SIRCFC = \{sircfc_j\}_{j=1}^6$  – множество свойств классификационного признака отдельного требования ПО программного обеспечения (SIRCFC – Software Individual Requirement Classification Feature Characteristics),  $sircfc$  – свойство классификационного признака отдельного требования программного обеспечения,  $j$  – порядковый номер свойства классификационного признака отдельного требования программного обеспечения.

### 1.3. Атрибуты отдельного требования ПО

Атрибут – это неотъемлемый элемент сущности, которая может быть количественно или качественно установлена экспертом или средствами автоматизации [12]. В данном случае сущностью является отдельное требование ПО.

Рассмотрим множество потенциальных атрибутов для отдельного требования ПО:

1. Идентификация (identification). Каждое требование должно быть однозначно идентифицировано (т.е. уникальный номер, тег, мнемоника). Идентификация, если это необходимо, может также включать связи и отношения с данным требованием;

2. Приоритет заинтересованных сторон (stakeholder priority). Для каждого требования должен быть определен приоритет. Он может быть установлен на основе консенсуса между потенциальными заинтересованными сторонами. При необходимости для определения приоритета каждого требования может применяться такие шкалы, как: от 1 до 5; высокая, средняя или низкая;

3. Зависимость (dependency). Требования могут зависеть друг от друга. Если такая зависимость существует, то она должна быть определена. Некоторые требования могут иметь низкий приоритет с точки зрения одной из заинтересованных сторон, но, тем не менее, имеют важное значение для реализации программного обеспечения, поскольку они связаны с требованиями с более высоким приоритетом;

4. Риск (risk). Методы анализа риска могут использоваться для определения соответствия системных требований с существующими рисками и тяжестью их последствий;

5. Источник (source). Каждое требование должно включать атрибут, который указывает источник для требования. Для отдельных требований могут быть определены несколько источников;

6. Обоснование (rationale). Каждое требование должно содержать обоснование. Обоснование со-

держит причину необходимости выполнения требования и указывает на результаты анализа, исследования, моделирования или другие существенные объективные доказательства;

7. Сложность (difficulty). Для требования целесообразно определить предполагаемую сложность его выполнении (например, легко / номинально / сложно). Такая дополнительная информация дает возможность проанализировать требования с точки зрения их выполнения и моделирования затрат;

8. Тип (type). Требования различаются по назначению и по типам свойств, которые они представляют. Это помогает их классифицировать в отдельные логические группы. В качестве типов требований будем рассматривать функциональные и нефункциональные;

9. Номер версии (version number). Указание версии требования. Версия требования необходима для того, чтобы убедиться, что реализуется правильная версия требования, а также для указания изменчивости требования.

Предварительный анализ представленных атрибутов для отдельного требования ПО позволил сформировать их множество, исключив атрибут «зависимость» (dependency), поскольку он не применим для отдельного требования ПО. Такой атрибут может характеризовать только профиль требований. Таким образом в множество атрибутов для отдельного требования ПО вошли следующие атрибуты: 1. «Идентификация» (identification), 2. «Приоритет заинтересованных сторон» (stakeholder priority), 3. «Риск» (risk), 4. «Источник» (source), 5. «Обоснование» (rationale), 6. «Сложность» (difficulty) и 7. «Тип» (type), 8. «Номер версии» (version number).

Таким образом каждое отдельное требование может обладать соответствующими 8 атрибутами. Введем формальные обозначения для таких атрибутов:

$$SIRA = \{sira_j\}_{j=1}^8 \text{ – множество атрибутов отдельного требования программного обеспечения}$$

(SIRA – Software Individual Requirement Attributes),  $sira$  – атрибут отдельного требования программного обеспечения,  $j$  – порядковый номер атрибута отдельного требования программного обеспечения.

Рассмотрим следующие потенциальные атрибуты для классификационного признака отдельного требования ПО:

– идентификация (identification) – каждый классификационный признак должен быть однозначно идентифицирован, т.е. ему должен быть присвоен уникальный номер, тег или мнемоника;

– зависимость (dependency) – классификационные признаки могут зависеть друг от друга. Если

такая зависимость существует, то она должна быть определена. Зависимость классификационных признаков существует, если тип таксономической структуры иерархический или смешанный.

Из 2 представленных атрибутов каждому классификационному признаку отдельного требования может соответствовать 1 атрибут «идентификация» (identification), а атрибут «зависимость» (dependency) соответствует только классификационному признаку в профиле требований ПО. Введем формальные обозначения для таких атрибутов:

$$- \text{SIRCFA} = \{\text{sircfa}_j\}_{j=1}^1 - \text{множество атрибутов}$$

классификационного признака отдельного требования программного обеспечения (SIRCFA – Software Individual Requirement Classification Feature Attributes), sircfa – атрибут классификационного признака отдельного требования программного обеспечения, j – порядковый номер атрибута классификационного признака отдельного требования программного обеспечения.

#### 1.4. Требования к семантике отдельного требования ПО

Семантические конструкции для каждого требования ПО могут быть логически разбиты на следующие группы: обязательные, допустимые и не желательные. Рассмотрим каждую группу семантических конструкций более подробно:

– группа обязательных семантических конструкций содержит в себе наиболее важные семантические элементы требования. В качестве примера приведем наиболее распространённый и обязательный элемент для требования «должен» (shall);

– группа допустимых семантических конструкций содержит в себе элементы, которые не противоречат смыслу самого требования и могут быть использованы в нем. Примерами такой группы являются «следует» (should), «может» (may) и др.;

– группа не желательных семантических конструкций – это элементы требований, которые могут являться источниками неоднозначности трактовки требования, искажении его смысла, не полноты требования, его не точности и др. Примерами для такой группы могут являться следующие варианты: употребление слов в превосходной степени – «лучший» (best), «большинство» (most); субъективные словосочетания – «удобный для пользователя» (such as user friendly), «простой в использовании» (easy to use); неоднозначные словосочетания – «почти всегда» (almost always), «значительный» (significant), «минимальный» (minimal); не поддающиеся проверке словосочетания – «оказывать поддержку» (provide support), «но не ограничиваясь» (but not

limited to), «как минимум» (as a minimum), «лучше чем» (better than); своего рода ухищрения «если возможно» (if possible), «при необходимости» (as appropriate), «применимо» (as applicable); неполные ссылки (без указания ссылки с ее датой и номером версии, указания только применимых частей ссылки); отрицательные утверждения – «не должен» (shall not), «может не быть» (may not be), «может не иметь» (may not have).

Введем следующие обозначения для представленных групп:

$$- \text{MSC} = \{\text{msc}_i\}_{i=1}^n - \text{множество обязательных}$$

семантических конструкций (mandatory semantic structures). Заглавными буквами будет обозначено название множества (MSC), а буквами нижнего регистра элементы множества (msc), i – порядковый номер обязательной семантической конструкции;

$$- \text{ASC} = \{\text{asc}_i\}_{i=1}^n - \text{множество допустимых}$$

семантических конструкций (admissible semantic structures). Заглавными буквами будет обозначено название множества (ASC), а буквами нижнего регистра элементы множества (asc), i – порядковый номер допустимой семантической конструкции;

$$- \text{USC} = \{\text{usc}_i\}_{i=1}^n - \text{множество не желательных}$$

семантических конструкций (undesirable semantic structures). Заглавными буквами будет обозначено название множества (USC), а буквами нижнего регистра элементы множества (usc), i – порядковый номер не желательной семантической конструкции.

Общий анализ семантики классификационных признаков отдельных требований ПО показал, что специализированные требования в части их семантики отсутствуют. Таким образом, отсутствие требований для семантики можно записать следующим образом:

–  $\text{SIRCF SR} = \emptyset$  – множество требований к семантике классификационного признака отдельного требования программного обеспечения (SIRCF SR – Software Individual Requirement Classification Feature Semantic Requirements).

#### 1.5. Требования к синтаксису отдельного требования ПО

Каждое требование состоит из одного и более предложений. Рассмотрим и сформулируем требования к структуре предложений, входящих в отдельное требование ПО. Проанализировав несколько источников [12, 14], в которых рассматриваются возможные варианты синтаксических структур требований, было принято решение сформировать максимально общую синтаксическую структуру для отдельного требования ПО. Полученная таким обра-

зом обобщенная семантическая структура отдельного требования ПО (рис. 3) состоит из 6 элементов. Представим более подробное описание каждого элемента обозначенной семантической структуры:

1. Условие – условие, которое обычно указывается в начале требования. Условие может начинаться со следующих слов: «когда», «если», «пока что», «где», «который».

2. Тема – тема самого требования, то есть описание, для чего предназначено такое требование. Примерами для темы могут быть: информационная система, программное обеспечение, автоматизированная система и др.

3. Действие – действия требования. Т.е. это действие, которое должно быть выполнено в рамках соответствующего требования. В качестве примера приведем следующие: установить, отключить, переключить, обнулить, сформировать, проверить, запустить и др.

4. Объект – объект требования, т.е. непосредственный объект для которого предназначено требование, например, показатель, сигнал, несущая, переключатель и др.

5. Ограничения – ограничение, которое является важным при выполнении указанного действия. Стоит отметить, что ограничения обычно применяются совместно с определенным условием. Например, с учетом, если установлен, пока осуществляется и др.

6. Значения – это значение, которое должен принимать объект при выполнении требования. Например, да или нет, 0 или 1, [0 ... 1], on или off.

Таким образом, требование в части своей структуры представляет собой множество из 6 возможных элементов. Ведем обозначения для множества элементов структуры требования:

–  $RSSE = \{rsse_i\}_{i=1}^n$  – множество элементов синтаксической структуры требования (Requirement Syntactical Structure Elements). Заглавными буквами будет обозначено название множества (RSSE), а буквами нижнего регистра элементы множества (rsse). Элементы множества RSSE представлены в

виде картежа, поскольку их последовательность важна и не должна нарушаться.

Количество элементов структуры требования (или мощность множества RSSE) не является постоянным. В связи с этим, очевидно, что каждое требование может включать разное количество элементов семантической структуры требования, т.е. количество структурных элементов может меняться в зависимости от требования. Сформируем множество возможных вариантов семантической структуры требования с учетом переменного количества ее элементов и представим их в более формализованном виде (табл. 2). В процессе формирования множества вариантов структуры требования были определены следующие особенности:

– элемент множества  $rsse_5$  (Ограничения) может проявляться в семантической структуре в двух вариантах: ограничения с условием и без условия. В связи с этим, для идентификации таких вариантов был введен дополнительный под индекс и варианты были обозначены следующим образом: ограничения с условием –  $rsse_{5,1}$ , ограничения без условия –  $rsse_{5,2}$ ;

– были определены элементы множества структуры требования, которые присутствуют во всех вариантах, т.е. являются постоянными. Это элементы  $rsse_2$  (тема) и  $rsse_3$  (действие), которые в табл. 2 выделены серым цветом.

Таким образом все элементы синтаксической структуры разделим на 2 группы:

–  $RSSEP = \{rssep_i\}_{i=1}^n$  – множество постоянных элементов синтаксической структуры требования (Requirement Syntactical Structure Elements Permanent);

–  $RSSEN = \{rsse_n\}_{i=1}^n$  – множество не постоянных элементов синтаксической структуры требования (Requirement Syntactical Structure Elements Not permanent).

Общий анализ синтаксиса классификационных признаков отдельных требований ПО показал, что специализированные требования в части их синтаксиса также отсутствуют.

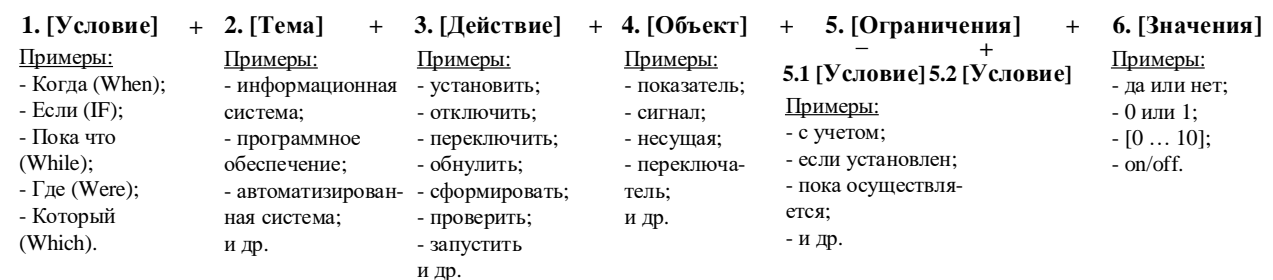


Рис. 3. Обобщенная семантическая структура отдельного требования ПО

Таблица 2

Варианты структур отдельного требования ПО

№ п/п	[Условие] (rsse <sub>1</sub> )	[Тема] (rsse <sub>2</sub> )	[Действие] (rsse <sub>3</sub> )	[Объект] (rsse <sub>4</sub> )	[Ограничения] (rsse <sub>5</sub> )		[Значения] (rsse <sub>6</sub> )	Формализованное описание вариантов структуры требования
					- [Условие] (rsse <sub>5.1</sub> )	+ [Условие] (rsse <sub>5.2</sub> )		
1.	+	+	+					RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> }
2.	+	+	+				+	RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>6</sub> }
3.	+	+	+		+			RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>5.1</sub> }
4.	+	+	+		+		+	RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>5.1</sub> , rsse <sub>6</sub> }
5.	+	+	+			+		RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>5.2</sub> }
6.	+	+	+			+	+	RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>5.2</sub> , rsse <sub>6</sub> }
7.	+	+	+	+				RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> }
8.	+	+	+	+			+	RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>6</sub> }
9.	+	+	+	+	+			RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.1</sub> }
10.	+	+	+	+	+		+	RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.1</sub> , rsse <sub>6</sub> }
11.	+	+	+	+		+		RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.2</sub> }
12.	+	+	+	+		+	+	RSSE = {rsse <sub>1</sub> , rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.2</sub> , rsse <sub>6</sub> }
13.		+	+					RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> }
14.		+	+				+	RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>6</sub> }
15.		+	+		+			RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>5.1</sub> }
16.		+	+		+		+	RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>5.1</sub> , rsse <sub>6</sub> }
17.		+	+			+		RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>5.2</sub> }
18.		+	+			+	+	RSE = {rse <sub>2</sub> , rse <sub>3</sub> , rse <sub>5.2</sub> , rse <sub>6</sub> }
19.		+	+	+				RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> }
20.		+	+	+			+	RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>6</sub> }
21.		+	+	+	+			RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.1</sub> }
22.		+	+	+	+		+	RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.1</sub> , rsse <sub>6</sub> }
23.		+	+	+		+		RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.2</sub> }
24.		+	+	+		+	+	RSSE = {rsse <sub>2</sub> , rsse <sub>3</sub> , rsse <sub>4</sub> , rsse <sub>5.2</sub> , rsse <sub>6</sub> }

Таким образом, отсутствие требований для синтаксиса классификационных признаков отдельных требований ПО можно записать следующим образом: SIRCFSYR = ∅ – множество требований к синтаксису классификационного признака отдельного требования программного обеспечения (SIRCFSYR – Software Individual Requirement Classification Feature SYntax Requirements).

## 2. Обобщенная структура модели качества отдельного требования ПО

Представим общую структуру модели отдельного требования ПО. Она включает в себя следующие элементы (рис. 4):

- структуру требования: множество элементов таксона, множество элементов классификационного признака;

- семантическую структуру отдельного требования ПО. Полная семантическая структура включает 6 таких элементов;

- множество свойств отдельного требования ПО – 19 свойств для каждого отдельного требования ПО и 6 свойств для классификационного признака;

- множество атрибутов отдельного требования ПО. Множество включает в себя 8 элементов для отдельного требования ПО и 1 атрибут для классификационного признака;

- множества семантических конструкций для отдельного требования ПО. Всего 3 множества: множество обязательных семантических конструкций, множество допустимых семантических конструкций,

- множество не желательных семантических конструкций.



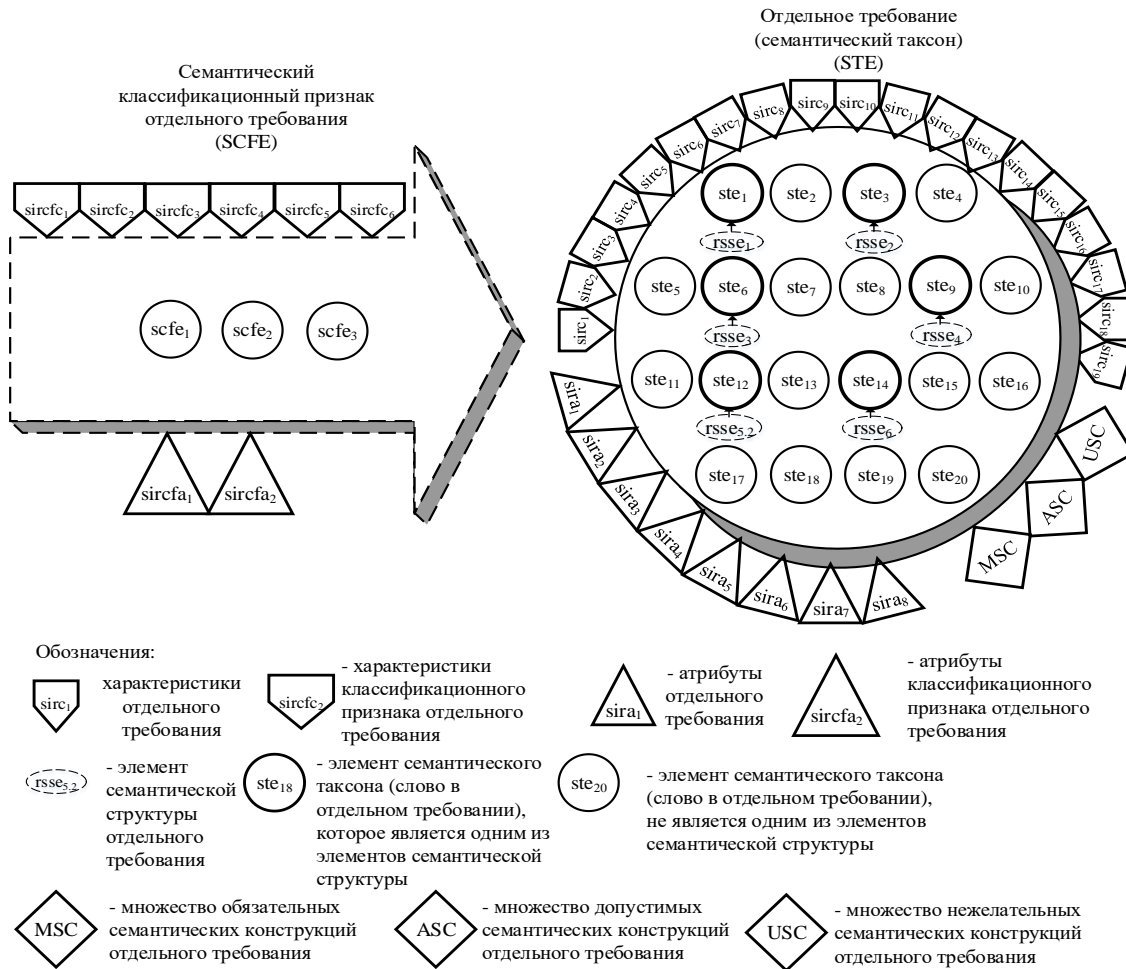


Рис. 4. Общая структура модели качества отдельного требования ПО

## Выводы

В статье предложена модель качества отдельного требования ПО и описаны все ключевые элементы такой модели. Модель состоит из пяти компонентов: структура отдельного требования ПО, свойства отдельного требования и его классификационного признака, атрибуты отдельного требования и его классификационного признака, требований к семантике отдельного требования ПО и требований к синтаксису отдельного требования. Также рассматриваются варианты структур отдельного требования ПО. Представленные в статье результаты в дальнейшем позволят разработать модель качества профиля требования ПО и модели оценки его качества.

## Литература

1. Knauss, E. *Assessing the Quality of Software Requirements Specifications [Text]* / E. Knauss, C. Boustani // *Requirements Engineering : Proc. of 16th IEEE Int. Conf., Barcelona, Catalunya, Spain, Septem-*

*ber 8-12, 2008. – Barcelona, 2008. – P. 341-343. DOI: 10.1109/RE.2008.29.*

2. Knauss, E. *Investigating the Impact of Software Requirements Specification Quality on Project Success [Text]* / E. Knauss, C. Boustani, T. Flohr // *Product-Focused Software Process Improvement : Proc. of the 10th Int. Conf. (PROFES), Oulu, Finland, June 15-17, 2009. – Berlin, 2009. – P. 28-42. DOI: 10.1007/978-3-642-02152-7\_4.*

3. Gasston, J. L. *A Quality Model for the Evaluation of Software Requirements Specifications [Text]* / J. Gasston // *Constructing the Infrastructure for the Knowledge Economy: Methods and Tools, Theory and Practice* / H. Linger et al. (eds). – Boston, 2004. – P. 281-291. DOI: 10.1007/978-1-4757-4852-9\_20

4. Gordieiev, O. *Usable Security Versus Secure Usability: an Assessment of Attributes Interaction [Text]* / O. Gordieiev, V. Kharchenko, K. Vereshchak // *ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer : Proc. of the 13th Int. Conf. (ICTERI), Kyiv, Ukraine, May 15-18, 2017. – K., 2017. – P. 727-740.*

5. Gordieiev, O. *Usability, security and safety interaction: profile and metrics based analysis [Text]* / O. Gordieiev, V. Kharchenko, K. Leontiiiev // *Dependability and Complex Systems : Proc. of the 13 Int. Conf. (DepCoS-RELCOMEX), Brunow, Poland, July 2-6,*

2018. – В., 2018. – P. 238-247. DOI: 10.1007/978-3-319-91446-6\_23.

6. Gordieiev, O. *Evolution of Software Quality Models: Green and Reliability Issues [Text]* / O. Gordieiev, V. Kharchenko, M. Fusani // *ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer : Proc. of the 11th Int. Conf. (ICTERI), Lviv, Ukraine, May 14, 2015. – Lviv, 2015. – P. 432-445.*

7. Gordieiev, O. *Evolution of software quality models: usability, security and greenness issues [Text]* / O. Gordieiev, V. Kharchenko, M. Fusani // *Computers : Proc. of the 19-th Int. Conf. (part of CSCC 15), Zakynthos Island, Greece, July 16-20, 2015. – Zakynthos Island., 2015. – P. 519-523.*

8. Gordieiev, O. *Software quality standards and models evolution: greenness and reliability issues [Text]* / O. Gordieiev, V. Kharchenko, M. Fusani // *Information and Communication Technologies in Education, Research, and Industrial Applications. – Cham, 2015. – P. 38-55.*

9. Kharchenko, V. *Profiling of Software Requirements for the Pharmaceutical Enterprise Manufacturing Execution System [Text]* / V. Kharchenko, O. Gordieiev, A. Fedoseeva // *Applications of Computational Intelligence in Biomedical Technology / R. Bris, J. Majernik, K. Pancarz, E. Zaitseva. – Springer, 2016. – Vol. 606. – P. 67-93.*

10. Андрашов, А. А. *Таксономические модели профилирования требований информационно-управляющих систем критического применения [Текст]* / А. А. Андрашов // *Радіоелектронні і комп'ютерні системи. – 2010. – № 7(48). – С. 104-108.*

11. *International standard ISO/IEC/IEEE 29148:2011(E). Systems and software engineering – Life cycle processes – Requirements engineering. – International Organization for Standardization, International Electrotechnical Commission, Institute of Electrical and Electronics Engineers, 2011. – 95 p.*

12. *International standard ISO/IEC/ IEEE 29148 :2018(E). Systems and software engineering – Life cycle processes – Requirements engineering. – International Organization for Standardization, International Electrotechnical Commission, Institute of Electrical and Electronics Engineers, 2018. – 104 p.*

13. *International standard ISO/IEC 25012:2008. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model. – International Organization for Standardization, International Electrotechnical Commission, 2008. – 13 p.*

14. *Easy Approach to Requirements Syntax (EARS) [Text]* / A. Harwood, P. Wilkinson, A. Mavin, M. Novak // *Requirements Engineering : Proc. of the 17th IEEE Int. Conf., Atlanta, Georgia, USA, August 31-September 4, 2009. – Atlanta, 2009. – Vol. 1 – P. 317-322.*

## References

1. Knauss, E., Boustani, C. *Assessing the Quality of Software Requirements Specifications. In: 16th IEEE International Requirements Engineering Conference, 8-*

*12 September, 2008, Barcelona, Catalunya, Spain, pp. 341-343. DOI: 10.1109/RE.2008.29.*

2. Knauss, E., Boustani, C., Flohr, T. *Investigating the Impact of Software Requirements Specification Quality on Project Success. In: 10th International Product-Focused Software Process Improvement Conference, Oulu, Finland, June 15-17, 2009, pp. 28-42. DOI: 10.1007/978-3-642-02152-7\_4.*

3. Gasston, J. *Quality Model for the Evaluation of Software Requirements Specifications. In: Linger H. et al. (eds) Constructing the Infrastructure for the Knowledge Economy, 2004, pp. 281-291. DOI: 10.1007/978-1-4757-4852-9\_20*

4. Godieiev, O., Kharchenko, V., Vereshchak, K. *Usable Security Versus Secure Usability: an Assessment of Attributes Interaction. In: Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI), Kyiv, Ukraine, May 15-18, 2017, pp. 727-740.*

5. Gordieiev, O., Kharchenko, V., Leontiiiev, K. *Usability, security and safety interaction: profile and metrics based analysis. Proceedings of the 13 International Conference on Dependability and Complex Systems (DepCoS-RELCOMEX), Brunow, Poland, July 2-6, 2018, pp. 238-247. DOI: 10.1007/978-3-319-91446-6\_23.*

6. Gordieiev, O., Kharchenko, V., Fusani, M. *Evolution of Software Quality Models: Green and Reliability Issues. In: Proceedings of the 11th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer (ICTERI), Lviv, Ukraine, May 14, 2015, pp. 432-445.*

7. Gordieiev, O., Kharchenko, V., Fusani, M. *Evolution of software quality models: usability, security and greenness issues. In: Proceedings of the 19-th International Conference on Computers (part of CSCC 15), Zakynthos Island, Greece, July 16-20, 2015, pp. 519-523.*

8. Gordieiev, O., Kharchenko, V., Fusani, M. *Software Quality Standards and Models Evolution: Greenness and Reliability Issues. In: Information and Communication Technologies in Education, Research, and Industrial Applications. ICTERI 2015. Communications in Computer and Information Science, vol. 594, pp. 38-55.*

9. Kharchenko, V., Gordieiev, O., Fedoseeva, A. *Profiling of Software Requirements for the Pharmaceutical Enterprise Manufacturing Execution System. In: Bris, R., Majernik, J., Pancarz, K., Zaitseva, E. (eds) Applications of Computational Intelligence in Biomedical Technology. Studies in Computational Intelligence, vol. 606, pp. 67-93.*

10. Андрашов, А. А. *Таксономические модели профилирования требований информационно-управляющих систем критического применения [Taxonomic models of profiling requirements for critical information management systems]. Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems, 2010, no. 7(48), pp. 104-108.*

11. *International standard ISO/IEC/ IEEE 29148 :2011(E). Systems and software engineering – Life cycle processes – Requirements engineering, International*

Organization for Standardization, International Electrotechnical Commission, 2011. 95 p.

12. *International standard ISO/IEC/ IEEE 29148 :2018(E). Systems and software engineering – Life cycle processes – Requirements engineering*, International Organization for Standardization, International Electrotechnical Commission, 2018. 104 p.

13. *International standard ISO/IEC 25012:2008. Software engineering – Software product Quality Re-*

*quirements and Evaluation (SQuaRE) – Data quality model*, International Organization for Standardization, International Electrotechnical Commission, 2008. 13 p.

14. Harwood, A., Wilkinson, P., Mavin, A., Novak, M. Easy Approach to Requirements Syntax (EARS). In: *Proceeding. of the 17th IEEE International Requirements Engineering Conference*, August 31 – September 4, Atlanta, Georgia, USA, 2009, vol. 1, pp. 317-322.

Поступила в редакцію 12.03.2020, рассмотрена на редколлегии 15.04.2020

## МОДЕЛЬ ЯКОСТІ ОКРЕМОЇ ВИМОГИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

*О. О. Гордєєв*

Оснoву технічного завдання на розробку програмного забезпечення (ПЗ) становить профіль вимог, у якому враховуються функціональність, особливості, обмеження, ризики тощо майбутнього програмного забезпечення. Профіль вимог є продуктом процесу профілювання і являє собою таксономічну структуру, яка пов'язує між собою множину вимог щодо програмного забезпечення, яке розробляється. Неподільною одиницею профілю вимог є окрема вимога ПЗ. Формально вимога ПЗ являє собою набір пов'язаних пропозицій, а фактично є більш складним об'єктом, від якості якого залежить якість ПЗ в цілому. Імплементация в програмне забезпечення вимоги з недостатньою якістю тягне за собою ресурсні втрати. Існуючі роботи, пов'язані з цією проблематикою, у повному обсязі не пропонують представлення моделі якості окремої вимоги. Метою статті є розробка моделі якості окремої вимоги ПЗ. Об'єктом дослідження є вимога програмного забезпечення. Стаття присвячена розробці моделі якості окремої вимоги програмного забезпечення. Ідея розробки означеної моделі з'явилася після аналізу наступних стандартів: ISO / IEC / IEEE 29148: 2011 року (E), ISO / IEC / IEEE 29148: 2018 (E) і ISO / IEC 25012: 2008. Положення, які представлені в позначених стандартах і стали основою даної статті. У ній розглядається вимога, як окремий непов'язаний елемент профілю вимог програмного забезпечення. Вимога подається у вигляді елементів фасетного-ієрархічних структур і складається з семантичної класифікаційної ознаки і семантичного таксона. Запропонована п'ятикомпонентна модель якості окремої вимоги ПЗ, яка включає структуру, властивості, атрибути, вимоги до синтаксису та семантики. Об'єднання таких елементів в одній моделі дозволяє формально описати якість окремої вимоги ПЗ. Як результат, в даній роботі пропонується формальний опис і представлення моделі якості окремої вимоги програмного забезпечення.

**Ключові слова:** вимога ПЗ; профіль вимог ПЗ; якість окремої вимоги ПЗ; фасетно-ієрархічна структура; модель якості ПЗ; властивість вимоги ПЗ.

## SOFTWARE INDIVIDUAL REQUIREMENT QUALITY MODEL

*O. Gordieiev*

The basis of the specification for software development is the requirements profile, which takes into account functionality, features, limitations, risks, etc. of future software. The requirements profile is a product of the profiling process and is a taxonomic structure that links together many of the requirements for the software being developed. An indivisible unit of the requirements profile is a separate software requirement. Formally, the software requirement is a set of related requirements, but it is a more complex object, the quality of the software as a whole depends on its quality. The implementation of insufficient quality requirements in software entails resource losses. Existing works related to this issue do not fully propose the presentation of the quality model of a particular requirement. The purpose of this article is to develop a quality model of an individual software requirement. The object of research is the software requirement. The article is devoted to the development of a quality model of an individual software requirement. The idea of developing the designated model came about after analyzing the following standards: ISO / IEC / IEEE 29148: 2011 (E), ISO / IEC / IEEE 29148: 2018 (E) and ISO / IEC 25012: 2008. The provisions that are presented in the designated standards and formed the basis of this article. It considers the requirement as a separate, unrelated element of the software requirements profile. The requirement is represented in the form of elements of facet-hierarchical structures and consists of a semantic classification attribute and a semantic taxon. A five-component model of the quality of an individual software requirement is proposed, it includes structure, properties, attributes, syntax, and semantics requirements. The combination of such elements in one model allows us to formally describe the quality of an individual software requirement. As a result, this article proposes a formal description and presentation of the quality model of an individual software requirement.

**Keywords:** software requirement; software requirements profile; quality of individual software requirement; facet-hierarchical structure; software quality model; software requirements property.

**Гордєєв Александр Александрович** – канд. техн. наук, доцент, заведуючий кафедри кібербезпеки ГВУЗ «Університет банківського дела», Київ, Україна, e-mail: alex.gordeyev@gmail.com.

**Oleksandr Gordieiev** – PhD, associate professor, head of cybersecurity department at Banking University, Kyiv, Ukraine, e-mail: alex.gordeyev@gmail.com, Scopus Author ID: 56447150300, ORCID Author ID: 0000-0003-2517-9388, ResearchGate: Oleksandr\_Gordieiev.