

DOI: 10.15514/ISPRAS-2023-35(6)-2



Усовершенствование модели оценки нефункциональных требований, классифицирующей диапазоны одинакового размера с помощью алгоритма k-ближайших соседей

¹ Ф. Вальдес-Соуто, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>

² Х. Валериано-Ассем, ORCID: 0009-0008-6473-1271 <jorge.valeriano@spingere.com.mx>

¹ Д. Торрес-Робледо, ORCID: 0009-0002-7168-9709 <dtorres@ciencias.unam.mx>

¹ Национальный автономный университет Мексики, факультет Науки, Мехико, Мексика.
² SPINGERE, Мехико, Мексика.

Аннотация. Любой проект разработки программного обеспечения должен оценивать нефункциональные требования. Обычно для таких оценок руководители разработок программного обеспечения вынуждены обращаться к экспертам. Сегодня не существует стандартизированных программных инструментов для оценки нефункциональных требований, поэтому большинство моделей оценки сосредоточены на изучении функциональных требований пользователя и не учитывают нефункциональных, хотя оба эти термина часто являются субъективными. Целью этой статьи было показать, как, применяя нечеткую логику и алгоритм k-ближайших соседей, в мексиканской компании для решения некоторой конкретной проблемы была построена модель оценки нефункциональных требований, учитывающая указанную субъективность терминологии. Предложенная модель использовала базы данных реальных проектов этой частной мексиканской компании.

Ключевые слова: стандарт COSMIC, функциональные точки COSMIC, нефункциональные требования, функциональные требования, машинное обучение, KNN-классификация, оценка затрат, модель оценки состояния проектов в условиях неопределенности EPCU.

Для цитирования: Вальдес-Соуто Ф., Валериано-Ассем Х., Торрес-Робледо Д. Усовершенствование модели оценки нефункциональных требований, классифицирующей диапазоны одинакового размера с помощью алгоритма k-ближайших соседей. Труды ИСП РАН, том 35, вып. 6, 2023 г., стр. 29–42. DOI: 10.15514/ISPRAS-2023-35(6)-2.

Improving a Model for NFR Estimation Classifying Equal Size Bands with KNN

¹ F. Valdés-Souto, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>

² J. Valeriano-Assem, ORCID: 0009-0008-6473-1271 <jorge.valeriano@spingere.com.mx>

¹ D. Torres-Robledo, ORCID: 0009-0002-7168-9709 <dtorres@ciencias.unam.mx>

¹ National Autonomous University of Mexico Science Faculty, CDMX, México.

² SPINGERE, CDMX, México.

Abstract. Any software development project needs to estimate Non-Functional Requirements (NFR). Typically, software managers are forced to use expert judgment to estimate the NFR. Today, NFRs cannot be measured, as there is no standardized unit of measurement for them. Consequently, most estimation models focus on the Functional User Requirements (FUR) and do not consider the NFR in the estimation process because these terms are often subjective. The objective of this paper was to show how an NFR estimation model was created using fuzzy logic, and K-Nearest Neighbors classifier algorithm, aiming to consider the subjectivity embedded in NFR terms to solve a specific problem in a Mexican company. The proposed model was developed using a database with real projects from a Mexican company in the private sector.

Keywords: COSMIC, CFP, NFR, FUR, ML, KNN classification, Effort estimation, EPCU.

For citation: Valdés-Souto F., Valeriano-Assem J., Torres-Robledo D. Improving a model for NFR estimation classifying equal size bands with KNN. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 6, 2023. pp. 29-42 (in Russian). DOI: 10.15514/ISPRAS-2023-35(6)-2.

1. Введение

Оценка программного обеспечения привлекает внимание многочисленных исследователей с момента его появления в 1950-х годах и в течение более чем 70-летнего существования [1]. Эти исследователи обнаружили среди прочего, что правильная оценка важна для успеха разработки и оказывает значительное влияние на бюджетирование и планирование проектов в отрасли [2-4].

В литературе по оценке программного обеспечения за более чем шесть десятилетий был разработан широкий спектр методов оценки [5-6], множество классифицирующих методов оценки [7, 2, 4, 8-9] и топологии процесса оценки [10-11], однако, до сих пор не достигнуто соглашение по единой модели, которая устойчиво дает точные результаты для всех производственных проектов. Модели оценки тесно связаны с процессом измерения входных переменных, используемых для генерации оценки [12]. В результате рекомендуется использовать стандартизированные метрики [13].

Однако даже когда функциональный размер (количество функциональных возможностей) является центральным элементом оценки, полный объем программного проекта определяется, по крайней мере, функциональными (Functional User Requirements, FUR) и нефункциональными (Nonfunctional Requirements, NFR) требованиями пользователя. Вот почему некоторые авторы указывают, что с учетом NFR результат оценки улучшается, хотя эти элементы рассматриваются в очень немногих исследованиях [14].

При использовании NFR наблюдаются и некоторые проблемы, в частности, отсутствие консенсусной терминологии, что приводит к значительному разнообразию терминов NFR [15-16], а также к их субъективному пониманию [17].

В этой статье представлен практический подход, разработанный и примененный в мексиканской организации, которая использует для измерения функционального размера стандартную методологию консорциума COSMIC (COmmon Software Measurement International Consortium), хотя полученные оценки имеют высокую дисперсию. Проанализировав основную причину разброса, разработчики пришли к выводу, что она

заключается в разнообразии нефункциональных требований. Следовательно, компания должна количественно оценить NFR, чтобы оценить необходимые усилия по проекту.

Настоящая работа представлена по следующей схеме. В разделе 2 приведен краткий обзор моделей оценки состояния проектов в условиях неопределенности (Estimation of Projects in Contexts of Uncertainty, EPCU), классификатора для машинного обучения, использующего алгоритм k-ближайших соседей (K-Nearest Neighbors, KNN), методов классификации диапазонов равного размера и показывается необходимость измерения или оценки NFR. В разделе 3 показаны этапы разработки модели оценки усилий на реализацию NFR с описанием используемых методов анализа и алгоритмов. В разделе 4 представлены выводы, сделанные на основе проведенных исследований. В разделе 5 описываются направления будущих исследований и выявленные ограничения предлагаемой модели оценки.

2. Обзор литературы

2.1 Назначение программного обеспечения

Требования к проекту программной системы, названные Бульони “Требованиями пользователя” (User Requirements, UR), делятся на две группы [18]. Первая группа требований состоит из функциональных (связанных с реализуемым проектом) и нефункциональных (связанных с реализуемым продуктом) требований, вторая группа – требований относится к некоторым иным потенциальным результатам, связанным с проектом. Руководством по управлению проектами (Project Management Body of Knowledge, PMBOK), даются два определения: рамки или границы продукта и рамки проекта [19]. В этих двух подходах можно наблюдать прямую связь между требованиями к проекту и рамками продукта, а также между требованиями к продукту и рамками проекта.

Особый подход предложен стандартом COSMIC, точнее ISO/IEC 19761:2011 [16], где требования к проекту программной системы разделены на функциональные, нефункциональные, а также на требования и ограничения проекта (Project Requirements and Constraints, PRC).

2.2 Нефункциональные требования (NFR)

“Стандарт” для “измерения” NFR был опубликован в 2019 году под названием IEEE 2430™ (<https://standards.ieee.org/ieee/2430/7045/>) на основе рекомендаций Международной группы по изучению функциональных точек IFPUG (International Function Point Users Group), которая предложила свой подход к измерению нефункциональных требований к программному обеспечению [20], получивший наименование “процесс нефункциональной оценки программ” (Software Non-functional Assessment Process, SNAP). Такие функциональные точки получили наименование функциональных точек COSMIC (COSMIC functional points, CFP). Стандарт 2019 года является примером ошибочной метрологической практики, что делает его бесполезным: он не позволяет сравнивать разные проекты [21].

Терминология нефункциональных требований столь обширна, что, согласно [16], “мы далеки от полного, общепринятого списка” терминов NFR, большинство из них субъективны. Это затрудняет их оценку и придание им численных значений [17].

В работе [16] был представлен глоссарий NFR, включающий 60 терминов, разделенных на три основных класса: требования к качеству (систем программного обеспечения), требования к системной среде и технические требования. Этот глоссарий основан на нескольких документах, таких как [22-24, 14-15]. Важно учитывать, что некоторая часть нефункциональных требований в течение жизненного цикла проекта может быть проявлена в виде требований функциональных, а этот цикл можно измерять с помощью какого-либо стандартного метода, например, COSMIC.

2.3 Оценка программного обеспечения с учетом нефункциональных требований

Хотя оценки необходимых на разработку программ усилий существенно влияют на планирование проектов и составление их бюджетов, методы получения таких оценок все еще остаются предметом дискуссий [4]. Принято считать, что любая модель оценки тесно связана с процессом измерения входных переменных, используемых для генерации оценки [1]. Причина в том, что функциональный размер – единственная характеристика, которую можно последовательно оценить количественно [13].

Однако для создания моделей оценки, кроме функционального размера, используются и другие переменные. Функциональный размер – не единственная характеристика стоимости или трудозатрат, необходимых для реализации программного проекта [6]. Сравнительное исследование, проведенное в работе [4], показало, что “большинство исследований (71,67%) используют несколько факторов затрат, а не отдают приоритет конкретному”. Кроме того, Джонс в работе [25] рассматривает четыре важные характеристики, влияющие на методологию оценки программного обеспечения:

- опыт персонала,
- используемые технологии (языки программирования, инструменты поддержки и так далее),
- процесс разработки и
- среду программирования, в которой работает разработчик.

Хотя всеми признано, что для улучшения оценки следует учитывать нефункциональные требования, согласно [14], только 33% изученных моделей оценки использовали нефункциональные требования в систематическом обзоре. Авторы при этом утверждают, что принятие во внимание таких требований снижает ошибку оценки на 30%.

Авторы работы [38] отмечают, что существуют отдельные нефункциональные требования, которые, пусть и в разной степени, прямо влияют на каждое функциональное требование. В работе [39] упоминается, что программное обеспечение со сложной архитектурой и связанные с ним системные нефункциональные требования не могут быть оценены с использованием единого коэффициента производительности для всех архитектурных компонентов.

Одной из основных проблем при использовании нефункциональных требований является отсутствие единого определения терминов [15-16], а ограничение, заключающееся в том, что большинство терминов нефункциональных требований являются субъективными, затрудняет последовательную оценку конкретного термина. Предложение рассматривать влияние конкретного нефункционального требования на каждое функциональное требование, которое рассматривается в работах [38-39], является более сложно реализуемым, чем рассмотрение его влияния на уровне проекта в целом. В настоящем исследовании мы использовали подход на уровне проекта.

Проведенное нами исследование было посвящено поискам способа включать нефункциональные требования в модель оценки таким образом, чтобы более явно проявлялось их влияние на оценки трудозатрат или бюджета. Оно должно быть полезно руководителям корпоративных проектов по разработке программного обеспечения.

2.4 Модель нечеткой логики для оценки субъективных переменных, EPCU

Модель оценки проектов в условиях неопределенности (EPCU), разработанная Валдисом и другими [26] в 2007 году, позже была использована не для оценки трудозатрат, а для

приближенного определения функционального размера [27-29]. В модели представлены шесть шагов:

1. идентификация входных переменных,
2. определение выходных переменных,
3. создание правил влияния,
4. фаззификация,
5. оценка правил влияния и
6. дефаззификация.

Первые три шага называются “контекстом EPCU”, который определяется как “набор переменных (входных и выходных) и отношений, которые влияют на конкретный проект или набор сходных проектов” [30].

В данной статье показывается, как использование в конкретном проекте трех основных классов NFR, определенных в работе [16], позволяет продемонстрировать достоинства подхода EPCU [26, 30-31] для характеристики нефункциональных требований.

2.5 Метод k-ближайших соседей

Алгоритм классификации, основанный на k-ближайших соседях (KNN), представляет собой контролируемую модель машинного обучения, целью которой является характеристика объекта в соответствии с его признаками. Он также стремится маркировать экземпляры в соответствии с расстояниями между значениями их признаков [32].

Важно отметить, что этот алгоритм лучше всего работает, если признаков у объектов достаточно много, поскольку их совокупность помогает лучше характеризовать объекты [33]. Выбор оптимального k зависит от данных, используемых для обучения. Выбор большого значения k уменьшает влияние шума в данных, однако, одновременно увеличивает вероятность перекрытия выявляемых классов [34].

2.6 Классификация групп равного размера

Метод определения функционального размера COSMIC (ISO/IEC 19761) вводит методы аппроксимации и используется, когда требования известны лишь частично, или определены на ранних стадиях проекта. Среди этих методов был введен метод диапазонов равного размера (Equal Size Bands, ESB).

В подходе выделения диапазонов равного размера [35] значения анализируемой переменной (функционального размера) делятся на несколько диапазонов. Все они имеют одинаковый размер в единицах, характерных для анализируемой переменной [35-36].

3. Практический пример

Проблема, с которой столкнулась компания, в которой мы проводили исследование, очень часто выявляется на различных предприятиях, разрабатывающих программное обеспечение. Наша работа выполнялась на основе исследования, описанного в работе [37], но с использованием наших собственных данных.

Компания уже пользовалась методологией COSMIC в течение последних семи лет для измерения функционального размера и применяет формальные модели оценки функциональных требований. Тем не менее полученные оценки характеризуются высокой дисперсией. Проанализировав основную причину разброса, специалисты пришли к выводу, что это связано с воздействием нефункциональных требований. Обычно производители

программного обеспечения вынуждены принимать во внимание экспертные заключения, основанные на субъективности NFR. Однако при таком подходе нефункциональные требования не могут выражаться количественно, и эксперты полагают, что результат оценивающей процедуры повторить невозможно. Отсюда следует, что компании нуждаются в методах количественных оценок нефункциональных требований и затрат на их реализацию. Наша компания регистрировала затраты на реализацию нефункциональных требований по реализуемым проектам, но не делала этого в разрезе каждого отдельного функционального требования. Таким образом, в данном тематическом исследовании нами использовался подход к оценке нефункциональных требований на уровне проекта, а не на уровне требований функциональных.

Вопрос, на поиски ответа на который направлено проведенное исследование, таков: *как оценить усилия, необходимые для разработки нефункциональных требований в проектах разработки программного обеспечения?*

Чтобы получить ответ на поставленный вопрос авторы использовали базу данных проектов компании. Релевантной информацией для анализа являлись идентификатор проекта и затраты на реализацию нефункциональных требований, выраженные в рабочих часах. В базе данных была обнаружена информация о 80 проектах, ранее выполненных компанией. Для разработки модели были использованы 57 проектов, остальные 23 проекта использовались для тестирования построенной модели.

3.1 Построение модели оценки нефункциональных требований (NFR)

Построение предлагаемой модели оценки нефункциональных требований выполняется в три этапа (рис. 1).

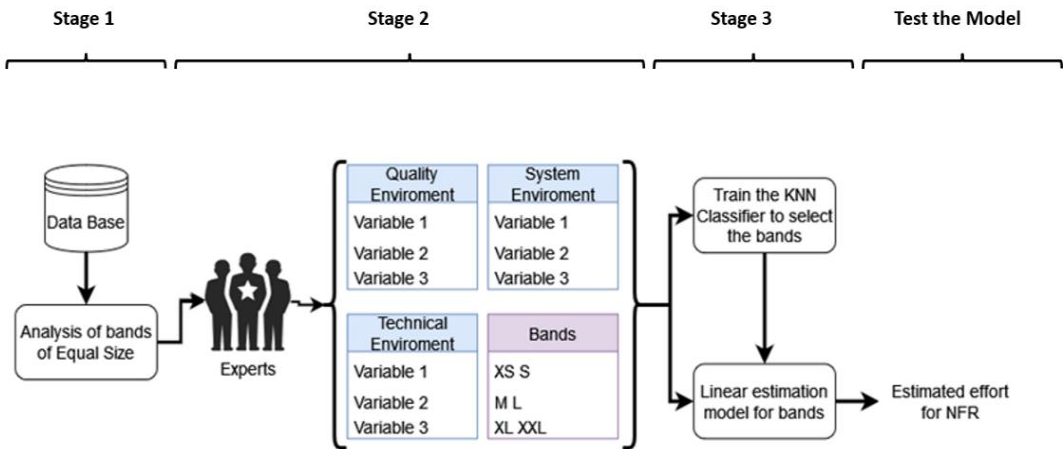


Рис. 1. Этапы разработки модели оценки затрат на нефункциональные требования
Fig. 1. Stages to develop the NFR effort estimation model

Первый этап заключается в отборе проектов, для которых реально выставлялись известные нефункциональные усилия. Анализ диапазонов равного размера (ESB) проводился на основе сведений о трудоемкости реализации требований, которая определялась в рабочих часах (WH). Этими классифицирующими диапазонами были выбраны следующие: XS (очень мало), S (мало), M (средне), L (много), XL (очень много), XXL (чрезвычайно много).

На втором этапе исследователи определили три контекста оценки проектов в условиях неопределенности EPCU. Группа экспертов заранее прошла обучение по оценке контекста EPCU и, приступив к работе, запросила входные переменные, связанные с оценкой индексов

NFR с использованием этих контекстов. При этом были получены результирующие количественные значения, соответствующие каждому контексту NFR и индексам NFR, то есть индексу требований к качеству (Quality Requirements Index, QRI), индексу требований к системной среде (System Environment Requirements Index, SERI) и индексу технических требований (Technical Requirements Index, TRI). Кроме того, экспертной группе было предложено приписать каждый проект к одному из диапазонов, ранее определенных с использованием экспертного заключения (табл. 1).

На третьем этапе выполняются два шага: во-первых, следует обучить классификатор KNN, во-вторых, разработать модель полилинейной регрессии с использованием индексов и с учетом выбранного диапазона в качестве входных переменных, при этом выходной переменной будут затраты, необходимые для реализации нефункциональных требований.

- a. **Классификатор KNN:** для обучения модели этот блок в качестве признаков использует индексы требований SERI, TRI и QRI, результатом классификации является выбираемый диапазон.
- b. **Линейный оценщик диапазона:** этот блок в качестве входных данных получает индексы SERI, TRI и QRI, а также диапазон, приписанный данному проекту классификатором KNN. Затем выбираются параметры, соответствующие выбранному диапазону: среднее значение, стандартное отклонение и сигма-фактор. Наконец, на основе трех влияющих индексов и параметров диапазона применяется многофакторная линейная модель, которая выдает значение оценки трудозатрат в рабочих часах.

Табл. 1. Второй этап оценки NFR
Table 1. Second stage for the NFR estimation

Проекты	Затраты на NFR (рабочие часы)	SERI	TRI	QRI	Диапазон, определённый экспертами
P1	1800	0.3	0.5	0.4	XL
P2	900	0.5	0.3	0.2	L
....					
Pn	440	0.6	0.4	0.3	M

3.2 Определение диапазонов равного размера

На первом этапе, чтобы определить количество диапазонов по исходной базе данных, выполнялась классификация групп одинакового размера в соответствии с трудозатратами на реализацию нефункциональных требований. В табл. 2 показаны результаты анализа с распределением проектов по шести диапазонам трудозатрат на реализацию нефункциональных требований. В табл. 3 показаны значения статистических параметров, которые использовались моделью оценки для каждого отдельного диапазона.

Табл. 2. Анализ групп одинакового размера по данным усилий NFR. μ = среднее значение, σ = стандартное отклонение, NA = неприменимо
Table 2. Equal size band analysis on the NFR effort data. μ = mean, σ = standard deviation, NA = Not Applicable

№ п/п	Размер (час)	Диапазон XS		Диапазон S		Диапазон M		Диапазон L		Диапазон XL		Диапазон XXL	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1	9094.7	159.6	203.8	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	4547.4	91.9	74.2	574.2	256.7	NA	NA	NA	NA	NA	NA	NA	NA
3	3031.6	70.4	55.4	277.5	90.9	771.8	208.5	NA	NA	NA	NA	NA	NA
4	2273.7	58.8	44.3	206.0	26.2	417.8	114.5	834.7	203.7	NA	NA	NA	NA
5	1818.9	51.0	36.4	178.0	22.9	279.9	61.0	556.0	62.1	952.1	17.1	NA	NA
6	1515.8	46.8	33.1	160.0	27.0	236.6	32.2	402.0	72.6	591.5	12.0	952.1	17.1

3.3 Определение контекстов для определения показателей влияния

Как ранее упоминалось, были определены три контекста EPCU, с помощью которых рассчитывалось влияние индексов каждой категории нефункциональных требований, определенных в работе [16]. Данные для определения контекстов приведены в табл. 4. Согласно концептуальной модели оценки NFR (рис. 1), выходные переменные – это числа в диапазоне от 0 до 1, отражающие степень влияния той или иной категории требований.

Табл. 3. Параметры, используемые моделью оценки
Table 3. Parameters used by the estimation model

Диапазон	Среднее значение (μ)	Стандартное отклонение (σ)	Используемый сигма-фактор
XS	46.8	33.1	1
S	160.0	27.0	3
M	236.6	32.2	3
L	402.0	72.6	3
XL	591.5	12.0	3
XXL	952.1	17.1	3

Табл. 4. Спецификация входных переменных контекста в соответствии с работой [16]
Table 4. Context's input variables specification [16]

Выходная переменная	Диапазон выходной переменной	Входная переменная	Диапазон входной переменной	Лингвистические группы		
Индекс качества среды (QRI)	[0, 1]	Сложность в зависимости от типа приложения (домена)	[0, 5]	Low	Mid	High
	[0, 1]	Сложность настройки сред разработки, тестирования и выполнения	[0, 5]	Low	Mid	High
	[0, 1]	Сложность от количества видов пользователей	[0, 5]	Low	Mid	High
Индекс требований к системной среде (SERI)	[0, 1]	Сложность системной производительности	[0, 5]	Low	Mid	High
	[0, 1]	Сложность данных, поддерживаемых системой	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за совместимости с другими системами	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за требуемого уровня легкости использования	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за требуемого уровня надежности	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за требуемого уровня управления доступом	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за требуемого уровня управления доступом	[0, 5]	Low	Mid	High
Индекс технических требований (TRI)	[0, 1]	Сложность операционной системы	[0, 5]	Low	Mid	High
	[0, 1]	Сложность базы данных	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за ограничений операционной платформы	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за требований разработки	[0, 5]	Low	Mid	High
	[0, 1]	Сложность проектирования интерфейсов	[0, 5]	Low	Mid	High
	[0, 1]	Сложность из-за выбранной архитектуры системы	[0, 5]	Low	Mid	High

3.4 Обучение классификатора KNN

Для обучения модели классификатора KNN мы будем использовать в качестве классифицирующих признаков три индекса, рассчитанные по рекомендациям EPCU (SERI, TRI и QRI), а метки для классификации будут задаваться диапазонами XS, S, M, L, XL, и XLL, которые были предложены группой экспертов.

После обучения классификатора мы получим модель, которая позволит классифицировать новые экземпляры по заданным параметрам (QRI, SERI, TRI).

3.5 Модель линейной оценки с использованием групп, выбранных моделью KNN

Согласно концептуальной модели оценки, показанной на рис. 1, модуль под названием “Линейная оценка в диапазонах” выполняет оценку трудозатрат на основе параметров:

- Влияющие индексы: QRI, SERI, TRI

- б. Диапазон, присвоенный проекту классификатором KNN
- с. Статистические параметры выбранного диапазона по результатам табл. 3

Вычисление оценки трудозатрат и длительности выполняется этим модулем решением линейного уравнения, основанного на индексах влияния (QRI, SERI, TRI) и параметрах выбранного диапазона. Получающееся значение может варьироваться от низкого предела до верхнего, что определяется интервалом $[\mu-\alpha\sigma, \mu+\alpha\sigma]$, где μ — среднее значение затрат в выбранном диапазоне, σ – стандартное отклонение, α – сигма-фактор. Все эти параметры показаны в табл. 3. При вычислениях используется следующее уравнение:

$$Затраты = f(QRI, SERI, TRI) = 2/3 (\alpha\sigma)(QRI + SERI + TRI) + \mu - (\alpha\sigma) \tag{1}$$

3.6 Результаты применения модели для оценки NFR

В итоге для проверки результатов модели оценки на соответствие реальным усилиям, требуемым для реализации NFR, мы выбрали 23 проекта. В табл. 5 приведены критерии качества, а результаты обработки набора тестовых данных приведены в табл. 6.

Видно, что более точные результаты получаются при использовании полной модели оценки NFR, которая использует индексы и автоматический выбор диапазонов (рис. 1). Результаты модели, которая использует индексы влияния только с использованием модели множественной линейной регрессии, несколько хуже.

Табл. 5. Критерии качества
Table 5. Quality criteria

	Предложенная полная модель оценки NFR (рис. 1)	Модель с использованием множественной линейной регрессией
Средняя величина относительной ошибки	107.8%	139.5%
Стандартное отклонение	136.5%	162.2%
Доля проектов с относительной ошибкой, меньшей 25%	26.1%	21.7%
Медианная величина относительной ошибки	47.4%	71.2%

Табл. 6. Набор тестовых данных
Table 6. Test data set

Проект	QRI	SERI	TRI	Прогноз диапазона	Фактический диапазон	Прогноз затрат	Фактические затраты
P01	0.18	0.13	0.3	M	M	179.04	153
P02	0	0.08	0.05	XS	XS	16.55	27
P03	0.2	0.16	0.21	M	M	176.47	185
P04	0.31	0.12	0.15	M	M	176.87	120
P05	0.34	0.22	0.37	L	M	200.02	65
P06	0.38	0.15	0.33	L	L	307.92	48
P07	0.15	0.11	0.15	S	XS	22.69	27
P08	0.12	0.2	0.07	M	S	100.06	27
P09	0.15	0.16	0.09	L	S	100.33	180

P10	0.2	0.26	0.08	L	M	174.78	36
P11	0.19	0.23	0.14	L	M	175.46	665
P12	0.12	0.07	0.06	XS	S	92.31	180
P13	0.39	0.36	0.33	M	L	340.03	200
P14	0.08	0.23	0.11	M	L	245.42	180
P15	0.14	0.4	0.21	M	L	293.39	540
P16	0.14	0.37	0.26	M	XXL	927.23	440
P17	0.62	0.79	0.43	M	M	258.78	320
P18	0	0.13	0	M	M	148.40	135
P19	0.18	0.32	0.18	L	L	282.65	500
P20	0.07	0.07	0.04	XS	S	88.58	27
P21	0.21	0.13	0.1	L	M	168.04	63
P22	0.05	0	0	S	S	81.72	55
P23	0.11	0.02	0.06	XS	S	89.52	85

4. Заключение

Модель оценки NFR, концепция которой проиллюстрирована на рис. 1, решает проблему оценки нефункциональных требований для организации, в которой проводилось исследование. Она допускает постоянное совершенствование, позволяющее повышать точность выполняемых оценок, поскольку она официально используется в организации и периодически перенастраивается.

Модель, разработанная для оценки NFR, является инновационной, поскольку она сочетает в себе широко известный в литературе набор нефункциональных требований с математическими элементами, которые позволяют эти требования оценить численно. В литературе нет описаний способов измерения подобных требований.

Используя информацию и модель, представленные в этой работе, можно предлагать локальные расширения метода измерения функционального размера COSMIC, вычисляя общий размер затрат на создание программных систем, включающий как функциональные, так и нефункциональные требования. При этом можно использовать следующее уравнение:

$$\text{Размер программной системы} = CFP + SERI + TRI + QRI + \text{Диапазон}$$

Эта информация помогает менеджерам определить влияние различных нефункциональных требований на конкретный проект и позволяет им генерировать количественную информацию для анализа этого влияния.

5. Будущие исследования

Чтобы подтвердить правильность предложенной модели оценки затрат на реализацию нефункциональных требований, мы намерены применить ее в других учреждениях.

Модель постоянно совершенствуется, это можно видеть на примере критериев качества, меняющихся по мере получения большего количества данных.

Использование различных найденных в литературе методов классификации, основанных на машинном обучении, позволит сравнить результаты и получить лучшее соответствие группам, описанным в этом тематическом исследовании.

Усовершенствовать модель можно, рассмотрев неоднородные архитектуры [39] и различные виды влияния нефункциональных требований на каждое из функциональных требований [38], сравнивая результаты с подходом на уровне проекта, предложенным в этой статье.

Список литературы / References

- O. Fedotova, L. Teixeira, A.H. Alvelos, Software effort estimation with multiple linear regression: Review and practical application, *J. Inf. Sci. Eng.* 29 (2013) 925–945.
- T.K. Lee, K.T. Wei, A.A.A. Ghani, Systematic literature review on effort estimation for Open Sources (OSS) web application development, in: *FTC 2016 - Proc. Futur. Technol. Conf.*, IEEE, San Francisco, California, USA, 2016: pp. 1158–1167. <https://doi.org/10.1109/FTC.2016.7821748>.
- P. Sharma, J. Singh, Systematic literature review on software effort estimation using machine learning approaches, in: *Proc. - 2017 Int. Conf. Next Gener. Comput. Inf. Syst. ICNGCIS 2017*, IEEE, Jammu, India, 2018: pp. 54–57. <https://doi.org/10.1109/ICNGCIS.2017.33>.
- C.E. Carbonera, K. Farias, V. Bischoff, Software development effort estimation: A systematic mapping study, *IET Res. Journals.* 14 (2020) 1–14. <https://doi.org/10.1049/iet-sen.2018.5334>.
- R. Silhavy, Z. Prokopova, P. Silhavy, Algorithmic optimization method for effort estimation, *Program. Comput. Softw.* 42 (2016) 161–166. <https://doi.org/10.1134/S0361768816030087>.
- M. Durán, R. Juárez-Ramírez, S. Jiménez, C. Tona, User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks, *Program. Comput. Softw.* 46 (2020) 569–583. <https://doi.org/10.1134/S0361768820080095>.
- M. Jørgensen, M. Shepperd, A systematic review of software development cost estimation studies, *IEEE Trans. Softw. Eng.* 33 (2007) 33–53. <https://doi.org/10.1109/TSE.2007.256943>.
- A. Abran, *Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers*, 1st ed., John Wiley & Sons, 2015.
- S. Bilgaiyan, S. Sagnika, S. Mishra, M. Das, A systematic review on software cost estimation in Agile Software Development, *J. Eng. Sci. Technol. Rev.* 10 (2017) 51–64. <https://doi.org/10.25103/jestr.104.08>.
- R. Britto, V. Freitas, E. Mendes, M. Usman, Effort estimation in global software development: A systematic literature review, *Proc. - 2014 IEEE 9th Int. Conf. Glob. Softw. Eng. ICGSE 2014*. (2014) 135–144. <https://doi.org/10.1109/ICGSE.2014.11>.
- F. Valdés-Souto, Validation of supplier estimates using cosmic method, *CEURInternational Work. Softw. Meas. Int. Conf. Softw. Process Prod. Meas. (IWSM Mensura 2019)*. 2476 (2019) 15–30.
- F. Valdés-Souto, L. Naranjo-Albarrán, Improving the Software Estimation Models Based on Functional Size through Validation of the Assumptions behind the Linear Regression and the Use of the Confidence Intervals When the Reference Database Presents a Wedge-Shape Form, *Program. Comput. Softw.* 47 (2021) 673–693. <https://doi.org/10.1134/S0361768821080259>.
- ISO/IEC, ISO/IEC 14143-1:2007 Information technology — Software measurement — Functional size measurement, (2007) 6. <https://www.iso.org/standard/38931.html>.
- S. Silva, M. Cortes, Use of Non-functional Requirements in Software Effort Estimation: Systematic Review and Experimental Results, *Proc. - 2017 5th Int. Conf. Softw. Eng. Res. Innov. CONISOFT 2017*. 2018-January (2018) 1–9. <https://doi.org/10.1109/CONISOFT.2017.00008>.
- European Cooperation for Space Standardization, *Space Engineering: Software- Part 1 Principles and Requirements*, (2005).
- Common Software Measurement International Consortium, *Guideline on Non-Functional & Project Requirements*, (2015).
- F. Valdés-Souto, A.S. Núñez-varela, H.G. Pérez-gonzález, Evaluating the software quality non-functional requirement through a fuzzy logic- based model based on the ISO / IEC 25000 (SQuaRE) standard, in: *2019 7th Int. Conf. Softw. Eng. Res. Innov.*, Conference Publishing Services (CPS), México, CDMX, 2019: pp. 16–25. <https://doi.org/10.1109/CONISOFT.2019.00014>.
- L. Buglione, The Next Frontier: Measuring and Evaluating Non-Functional Productivity, *Metr. Views, IFPUG Newsl.* 6 (2012) 11–14. <http://www.ifpug.org/Metric Views/MVBuglione.pdf>.
- Project Management Institute, *A Guide to the Project Management Body of Knowledge, PMBOK*, 5th ed., Project Management Institute, 2013.

- C. Tichenor, A new software metric to complement function points: The software non-functional assessment process (SNAP), *CrossTalk*. 26 (2013) 21–26.
- A. Abran, IEEE 2430 Non-Functional Sizing Measurements: A Numerical Placebo, *IEEE Softw.* 38 (2020) 113–120. <https://doi.org/10.1109/MS.2020.3028061>.
- P. Lago, P. Avgeriou, R. Hilliard, guest editors' introduction Software Architecture: *IEEE Softw.* (2010) 20–24.
- Y. Saito, A. Monden, K. Matsumoto, Evaluation of non-functional requirements in a request for proposal (RFP), in: *Proc. 2012 Jt. Conf. 22nd Int. Work. Softw. Meas. 2012 7th Int. Conf. Softw. Process Prod. Meas. IWSM-MENSURA 2012*, IEEE, 2012: pp. 106–111. <https://doi.org/10.1109/IWSM-MENSURA.2012.23>.
- L. Chung, B. Nixon, E. Yu, J. Mylopoulos, *Non-functional Requirements in Software Engineering*, Kluwer Academic Publishing, 2000.
- C. Jones, *Estimating Software Costs: Bringing Realism to Estimating*, Second, McGraw-Hill Companies, Inc., New York, N.Y., 2007.
- F. Valdés-Souto, A. Abran, Industry Case Studies of Estimation Models Using Fuzzy Sets, in: Reiner Dumke (Ed.), *Softw. Process Prod. Meas. Int. Conf. IWSM-Mensura 2007*, UIB-Universitat de les Illes Balears, Illes Balears, Spain, 2007: pp. 87–101.
- F. Valdés-Souto, A. Abran, Case Study: COSMIC Approximate Sizing Approach Without Using Historical Data, in: *Jt. Conf. 22nd Int. Work. Softw. Meas. 2012 Seventh Int. Conf. Softw. Process Prod. Meas., IEEE, Assisi, Italy*, 2012: pp. 178–189. <https://doi.org/10.1109/IWSM-MENSURA.2012.34>.
- F. Valdés-Souto, A. Abran, COSMIC Approximate Sizing Using a Fuzzy Logic Approach: A Quantitative Case Study with Industry Data, in: F. Vogelezang, M. Daneva (Eds.), *2014 Jt. Conf. Int. Work. Softw. Meas. Int. Conf. Softw. Process Prod. Meas., Conference Publishing Services (CPS), Rotterdam (Netherlands)*, 2014: pp. 282–292. <https://doi.org/10.1109/IWSM.Mensura.2014.44>.
- F.V. Souto, A. Abran, Improving the COSMIC approximate sizing using the fuzzy logic EPCU model, 2015. https://doi.org/10.1007/978-3-319-24285-9_13.
- F. Valdés-Souto, A. Abran, Comparing the Estimation Performance of the EPCU Model with the Expert Judgment Estimation Approach Using Data from Industry, in: R. Lee (Ed.), *Softw. Eng. Res. Manag. Appl. 2010*, Springer-Verlag, Berlin, 2010: pp. 227–240.
- F. Valdés-Souto, *Design of a Fuzzy Logic Software Estimation Process*, École De Technologie Supérieure, Université Du Québec, 2011.
- Jacob Goldberger, Sam Roweis, Geoff Hinton, Ruslan Salakhutdinov, Neighbourhood components analysis, *Adv. Neural Inf. Process. Syst.* 17 (2005) 513–520.
- T. Seidl, Nearest Neighbor Classification, *Encycl. Database Syst.* (2009). https://doi.org/https://doi.org/10.1007/978-0-387-39940-9_561.
- Scikit-Learn, 1.6. Nearest Neighbors, 2023. (n.d.). <https://scikit-learn.org/stable/modules/neighbors.html#classification>.
- A. Abran, A. Lestherhuis, B. Reynolds, A. Sellami, H. Soubra, S. Trudel, F. Valdés-Souto, F. Vogelezang, *Early Software Sizing with COSMIC: Experts Guide*, 2020 (2020) 1–67. <https://doi.org/10.13140/RG.2.1.4195.0567>.
- L. Lavazza, S. Morasca, Empirical evaluation and proposals for bands-based COSMIC early estimation methods, *Inf. Softw. Technol.* 109 (2019) 108–125. <https://doi.org/10.1016/j.infsof.2019.02.002>.
- B.R. Per Runeson, Martin Host, Austen Rainer, Case Study Research in Software Engineering: Guidelines and Examples, John Wiley & Sons, Inc., 2012. <https://doi.org/10.1002/9781118181034>.
- Fellir, F., Nafil, K., & Touahni, R. (2015). Analyzing the non-functional requirements to improve accuracy of software effort estimation through case-based reasoning. 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA). doi:10.1109/sita.2015.7358402.
- van der Vliet, Eric & Nijland, René & Mols, Harry & Vries, Jelle & Poort, Eltjo & Vogelezang, Frank. (2017). A Shortcut to Estimating Non-Functional Requirements? Architecture Driven Estimation as the Key to Good Cost Predictions. 10.1145/3143434.3143440.

Информация об авторах / Information about authors

Франсиско ВАЛЬДЕС-СОУТО - имеет докторскую степень в области инженерии программного обеспечения по специальности “Измерение и оценка программного обеспечения” в Высшей технологической школе (ETS) в Канаде, две магистерские степени в

Мексике и Франции. Президент COSMIC. Доцент факультета наук Национального автономного университета Мексики. Основатель Мексиканской ассоциации метрик программного обеспечения (AMMS). Более 25 лет опыта в разработке критически важного программного обеспечения. К настоящему времени опубликовал более 50 научных работ, включая статьи в индексируемых журналах, трудах научных конференций, книгах и главах книг. Является главным промоутером проекта изучения формальных метрик программного обеспечения в Мексике, продвигая COSMIC (ISO/IEC 19761) в качестве национального стандарта. Член Национальной системы исследователей (SNI). Область научных интересов: измерение и оценка программного обеспечения, применяемого для управления проектами программного обеспечения, управление тематикой, производительностью и экономикой разработок программного обеспечения.

Francisco VALDÉS-SOUTO – Doctor in Software Engineering with a specialty in Software Measurement and Estimation at the École de Technologie Supérieure (ETS) in Canada, two master's degrees in Mexico and France. President of COSMIC. Associate Professor of the Faculty of Sciences of the National Autonomous University of Mexico (UNAM). Founder of the Mexican Association of Software Metrics (AMMS). More than 25 years of experience in critical software development. He currently has more than 50 publications including articles in Indexed Journals, Proceedings, books and book chapters. He is the main promoter of the topic of formal software metrics in Mexico, promoting COSMIC (ISO/IEC 19761) as a National Standard. Member of the National System of Researchers (SNI). Research interests: software measurement and estimation applied to software project management, scope management, productivity and economics in software projects.

Хорхе ВАЛЕРИАНО-АСЕМ – Магистр компьютерных наук и инженерии в Национальном автономном университете Мексики, специалист-консультант по формальному измерению и оценке программного обеспечения с 2016 года. Сфера научных интересов: метрики программного обеспечения (COSMIC), модели оценки программного обеспечения, модели валидации программного обеспечения, оценка функциональных и нефункциональных требований, оценка эффективности проектов разработки программного обеспечения на основе метрик программного обеспечения, оценка качества программных продуктов.

Jorge VALERIANO-ASSEM - Master in Computer Science and Engineering from the National Autonomous University of Mexico, specialist consultant in formal software measurement and estimation since 2016. Areas of interest: Software metrics (COSMIC), Software estimation models, Software Validation Models, Estimation of Functional and Non-Functional Requirements, Evaluation of the Performance of Software Development Projects aligned to Software Metrics, Evaluation of the Quality of the Software Development Product.

Даниэль ТОРРЕС-РОБЛЕДО – магистрант Исследовательского института в области прикладной математики и систем, имеет ученую степень по программированию от научного факультета Национального автономного университета Мексики.

Daniel TORRES-ROBLEDO – Master student at Research Institute in Applied Mathematics and Systems, degree in Computer Science from Science Faculty of the UNAM.