DOI: 10.15514/ISPRAS–2020–33(2)–4

Оценка пользовательских историй на основе декомпозиции сложности с использованием байесовских сетей

[1] М. Дуран, ORCID: 0000-0002-6866-2647 <mayra.duran@uabc.edu.mx>
[1] Р. Хуарес-Рамирес, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>
[2] С. Хименес, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>
[1] К. Тона, ORCID: 0000-0003-4492-3432 <tona.caudia@uabc.edu.mx >

[1] Автономный университет Нижней Калифорнии (UABC),
Мексика, 21100, Нижняя Калифорния, Энсенада
[2] Тихуанский технологический институт,
Мексика, 22414, Нижняя Калифорния, Тихуана

Аннотация. На сегодняшний день в методологии разработки программного обеспечения Scrum предлагаются разные методы оценки трудозатрат и сложности пользовательских историй. В большинстве существующих методов факторы анализируются на уровне мелких структурных единиц, и эти методы не всегда точны. Хотя чаще всего для оценки пользовательских историй в Scrum используется покер планирования, он эффективен в основном для опытных команд, поскольку оценка строится на основе наблюдений экспертов, но у неопытных команд применение этого метода вызывает трудности. В данной статье мы предлагаем метод декомпозиции сложности на крупные блоки, позволяющий учитывать важные для оценки факторы. Для представления факторов и связей между ними используется байесовская сеть. Ребра взвешиваются на основе профессиональной оценки важности рассматриваемых факторов. Узлы сети представляют факторы. На этапе оценки Scrum-команда присваивает каждому фактору значение, что позволяет сети сгенерировать значения для сложности пользовательской истории с последующей трансформацией в номер карты покера планирования, которой представляет относительную оценку сложности пользовательской истории. Цель данного исследования состоит в том, чтобы предоставить командам разработчиков без опыта или без имеющихся статистических данных метод, позволяющий существенно повысить точность оценки сложности пользовательских историй с помощью модели, ориентированной на человеческий фактор разработчиков.

Ключевые слова: пользовательские истории; мелкоструктурная сложность; оценка; байесовская сеть; экспертное мнение.

Для цитирования: Дуран М., Хуарес-Рамирес Р., Хименес С., Тона К. Оценка пользовательских историй на основе декомпозиции сложности с использованием байесовских сетей. Труды ИСП РАН, том 33, вып. 2, 2021 г., стр. 77-92. DOI: 10.15514/ISPRAS–2021–33(2)–4.

Благодарности. Авторы выражают благодарность студентам программы по разработке программного обеспечения, обучавшимся в 2019 году в Автономном университете Нижней Калифорнии (UABC) и принявшим участие в Scrum-проектах, а также профессионалов этой области, представляющих различные компании региона. Благодаря вашему вкладу нам удалось собрать необходимые данные для успешного исследования. Мы также выражаем особую благодарность Национальному совету по науке и технологиям (CONACYT) за предоставленную стипендию для обучения на магистерской программе в области точных наук.

User Story Estimation based on the Complexity Decomposition using Bayesian Networks

[1] M. Durán, ORCID: 0000-0002-6866-2647 <mayra.duran@uabc.edu.mx>
[1] R. Juárez-Ramírez, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>
[2] S. Jiménez, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>
[1] C. Tona, ORCID: 0000-0003-4492-3432 <tona.caudia@uabc.edu.mx >

[1] Universidad Autónoma de Baja California,
Ensenada, Mexico, 21100
[2] Instituto Tecnológico de Tijuana, México,
Tijuana, Mexico, 22424

Abstract. Currently, in Scrum, there are different methods to estimate user stories in terms of effort or complexity. Most of the existing techniques consider factors in a fine grain level; these techniques are not always accurate. Although Planning Poker is the most used method in Scrum to estimate user stories, it is primarily effective in experienced teams since the estimation mostly depends on the observation of experts, but it is difficult when is used by inexperienced teams. In this paper, we present a proposal for complexity decomposition in a coarse grain level, in order to consider important factors for complexity estimation. We use a Bayesian network to represent those factors and their relations. The edges of the network are weighted with the judge of professional practitioners about the importance of the factors. The nodes of the network represent the factors. During the user estimation phase, the Scrum team members introduce the values for each factor; in this way, the network generates a value for the complexity of a User story, which is transformed in a Planning Poker card number, which represents the story points. The purpose of this research is to provide to development teams without experience or without historical data, a method to estimate the complexity of user stories through a model focused on the human aspects of developers.

Keywords: user stories; fine-grained complexity; estimation; Bayesian network; expert judge

For citation: Durán M., Juárez-Ramírez R., Jiménez S., Tona C. User Story Estimation based on the Complexity Decomposition using Bayesian Networks. Trudy ISP RAN/Proc. ISP RAS, vol. 33, issue 2, 2021, pp. 77-92 (in Russian). DOI: 10.15514/ISPRAS–2021–33(2)–4.

Acknowledgments. The authors would like to thank the students of Software Engineering course enrolled in 2019-1 and 2019-2 at Universidad Autónoma de Baja California who participated in the Scrum projects and the professionals in the area who participated from various companies in the region. Thanks to your participation we managed to collect data and information for the formation of our study. We also have special thanks to Consejo Nacional de Ciencia y Tecnología (CONACYT) for the scholarship to realize the master in science studies.

1. Введение

Scrum – это Agile-метод разработки, который определяется как итеративный, инкрементальный и эмпирический процесс, позволяющий управлять разработкой и осуществлять контроль над ней [1-3]. Это фреймворк, который представляет собой набор практических приемов для поддержки наглядного представления, экспертизы и адаптации проектов по разработке программного обеспечения [4-5]. Ключевым элементом Scrum являются пользовательские истории (user stories). Первая версия пользовательских историй – это краткое (одно или два предложения), простое и конкретное описание взаимодействия пользователя и разрабатываемого продукта [6]. Проект по разработке программного обеспечения включает в себя оценку стоимости, трудозатрат, размера и продолжительности [7].

При подсчете очков пользовательских историй (story points) следует учитывать множество факторов [8-11]. На качество и точность оценки напрямую влияют опыт и знания разработчиков, которые помогают определить и учесть все возможные факторы. По этой причине процесс оценки может быть проблематичным для начинающих разработчиков, так

как у них недостаточно опыта [12]. На оценку также негативно влияет неопределенность в требованиях [13]. В Scrum для оценки пользовательских историй можно использовать несколько методов, таких как покер планирования (Planning Poker) [8], Wideband Delphi [14], Fist of Five, Affinity Estimation [5] и T-Shirt Sizes [15-16].

Покер планирования – это наиболее часто используемый в Scrum метод для выполнения оценки пользовательских историй. Покер планирования – это метод, основанный на экспертном суждении, который заключается в присвоении очков каждой пользовательской истории для определения ее сложности [14]. Каждая пользовательская история оценивается в очках, которые отражают сложность ее реализации в рамках проекта. При применении покера планирования основной проблемой является высокая субъективность в оценке пользовательских историй. Поэтому покер планирования считается сложным и ненадежным методом, особенно для членов команды без опыта. Кроме того, оценка каждого отдельного члена команды является расплывчатой, поскольку основана на обобщенном представлении о сложности, в то время как действительности представление о сложности связано с рядом атрибутов: опытом работы с использованием данного языка программирования [17-18], опытом работы в предыдущих проектах [19-20], знанием текущего проекта [21], опытом использования технологий [17, 22], размером команды [23-24] и т.д. [14, 25-26].

Основная проблема существующих методов заключается в том, что в них сложность пользовательских историй рассматривается как единое целое [14, 25], однако некоторые авторы предлагают разделять сложность на несколько атрибутов [8, 13, 26-27] и учитывать каждый из них при оценке. В данной статье мы предлагаем рассмотреть набор атрибутов, влияющих на оценку сложности пользовательских историй. Часть атрибутов была сформулирована на основе опубликованного нами ранее систематического обзора литературы [28]. Окончательный набор атрибутов прошел экспертную проверку. Мы предлагаем использовать байесовскую сеть (БС), в который узлы представлены атрибутами, со взвешиванием соответствующих ребер на основе мнения практикующих профессионалов.

Мы использовали следующие критерии: опыт разработчика (опыт работы с технологиями, опыт работы с языком и опыт работы в предыдущих проектах), навыки разработчика, знание темы проекта и техническая сложность (функциональная зависимость, размер пользовательской истории (функции соединения, функции обработки данных)). Предложенная нами байесовская модель существенно облегчает процесс оценки. Во время оценки разработчики задают значения атрибутов, а байесовская модель вычисляет значение карты в очках.

Цель данного исследования состоит в том, чтобы предоставить командам разработчиков без опыта или накопленных данных метод, позволяющий существенно повысить точность оценки. Статья организована следующим образом. Разд. 2 содержит основные понятия. В разд. 3 представлены родственные работы. В разд. 4 описана методология исследования. В разд. 5 разбираются атрибуты, используемые в предлагаемой модели. Разд. 6 содержит информацию о процессе построения БС. В разд. 7 описано тестирование БС. В разд. 8 представлены результаты, в разд. 9 – выводы.

2. Основные понятия

2.1 Scrum

Scrum – самая популярная Agile-методология в индустрии программного обеспечения. Используя методы Scrum, многие компании улучшили качество результатов и продуктивность. Scrum – это Agile-метод разработки, который определяется как итеративный, инкрементный и эмпирический процесс управления проектом и контроля его выполнения [29-30]. В Scrum имеются три основные роли: ответственный за разработку продукта (Product Owner, PO), Scrum-мастер (Scrum Master, SM) и Scrum-команда (Scrum

Team, ST). PO является представителем заказчика. В Scrum имеются различные методы оценки, однако наиболее распространенным из них [8] является покер планирования.

2.2 Покер планирования

Покер планирования – это метод Scrum, использующийся для оценки командой разработчиков пользовательских историй с точки зрения трудозатрат. В игре фигурируют игральные карты с цифрами, которые основаны на модифицированной последовательности Фибоначчи (1, 2, 3, 5, 8, 13, 20, 40, 100). PO и все разработчики используют карты, чтобы достичь консенуса в оценке требований в журнале пожеланий продукта (Product Backlog) [7]. Цель покера планирования заключается в измерении сложности пользовательских историй в очках [14].

2.3 Пользовательские истории

Пользовательская история представляет собой короткое текстовое описание, отражающее только основные элементы стребования: для кого предназначен продукт, что ожидается от системы и, возможно, почему это важно [6]. Пользовательская история оценивается в очках, которые отражают сложность ее реализации [14][31].

2.4 Сложность

Когда мы говорим о сложности в контексте разработки программного обеспечения, речь может идти о сложности проблемы, на решение которой направлено разрабатываемое приложение, о структуре кода или об связях между элементами данных, которые будут обрабатываться в приложении. В случае пользовательских историй сложность кроется в различных элементах требуемой функциональности и аспектах их реализации. Определение сложности при оценке пользовательской истории – непростая задача, так как она включает в себя множество аспектов, эта проблема рассматривались в нескольких исследованиях об использовании БС, вероятностный характер которых позволяет нам работать с неопределенностью [10].

2.5 Байесовская сеть

БС относятся к семейству вероятностных графовых моделей. Эти графовые структуры используются для представления знаний о неопределенной области. Каждый узел в графе представляет собой случайную переменную, а ребра – вероятностные зависимости между соответствующими случайными переменными.

3. Родственные исследования

Для повышения качества оценок использовались различные методы. В этом разделе мы описываем работы, связанные с оценкой сложности и имеющие отношение к нашему предложению.

В своем исследовании С. Драгичевич (Srdjana Dragicevic) [10] и др. представляют модель БС, пригодную для прогнозирования трудозатрат. Она включает шесть атрибутов: тип задачи, сложность требований (сложность форм, сложность функций и сложность отчетов), качество спецификаций и навыки разработчика. Такой подход принес хорошие результаты. Однако используемые атрибуты сосредоточены на технических аспектах, которые может быть сложно распознать начинающим разработчикам.

Модель Д. Алостад (Jasem M. Alostad) и др. [11] основана на нечеткой логике, которая может улучшить оценку трудозатрат в Scrum. Авторы рассматривают четыре атрибута: опытность команды разработчиков, сложность задачи, размер задачи и точность оценки. В результате уровень точности удалось поднять до 60%. Однако в этой модели унализировались только

четыре элемента, тогда как для определения сложности пользовательских историй требуется больше атрибутов.

В работк Д. Мартинес (Janeth López-Martínez) и др. [8] предлагается модель БС для оценки пользовательских историй, основанная на экспертных знаниях. В этом исследовании использовались пять атрибутов – время, трудозатраты, опыт, приоритетность и значимость пользовательской истории, каждый из которых представляет узел в сети. Этот подход позволил получить хорошую корреляцию между традиционными оценками и оценками, полученными с помощью предложенной модели. Основной недостаток этой модели заключается в то, что авторы рассматривали опыт как единый фактор.

Как видно, в большинстве родственных работ учтены важные аспекты оценки, но при этом они обладают двумя существенными недостатками: 1) нет единого мнения о значимости каждого фактора и уровне детализации композиции; 2) никто детально не фокусировался на индивидуальных особенностях разработчиков на мелкоструктурном уровне.

4. Методология

Шаг 1. Выбор атрибутов модели: атрибуты были идентифицированы, проанализированы и отобраны для интеграции в предложенную модель. В разд. 5 показан процесс выбора атрибутов.

Шаг 2. Построение БС: БС была построена с использованием выбранного набора атрибутов, составляющих количественные и качественные компоненты. В разд. 6 подробно описывается, как была построена сеть.

Шаг 3. Тестирование модели: для тестирования модели были разработаны и проведены два различных эксперимента: один для студентов и один для профессионалов. В разд. 7 описана процедура, использованная в обоих экспериментах.

Шаг 4. Результаты и их валидация: в разд. 8 приведены полученные результаты. Полученные результаты были проанализированы и сопоставлены с результатами других исследований.

5. Предлагаемая модель байесовской сети

В этом разделе объясняется выбор атрибутов, предлагаемых для использования в модели. Был проведен анализ атрибутов и классификаций из систематического обзора литературы [28], в котором собрано более 50 атрибутов различных категорий, способных повлиять на оценку пользовательских историй. Основываясь на мнениях профессионалов, мы отобрали наиболее релевантные из них: опыт работы с технологиями, опыт работы с используемым языком программирования, опыт работы в предыдущих проектах, навыки разработчика, знание темы проекта.

Хотя основное внимание в нашей модели уделяется индивидуальным атрибутам, мы добавили несколько атрибутов, связанных с технической сложностью пользовательских историй. Речь идет о таких важных и часто используемых в оценке атрибутах, указывающих на техническую сложность пользовательской истории, как размер и зависимость. Выбранные атрибуты первого уровня описаны в табл. 1. Атрибуты были сгруппированы в узлы в зависимости от их природы. Окончательная модель показана на рис. 1.

Табл. 1. Выбранные атрибуты
Table 1. Selected attributes

Атрибут	Определение
Опыт работы с технологиями	Разработчик знаком с задействованными инструментами, методами и технологиями и уверенно их использует.
Опыт работы в предыдущих проектах	Знания и навыки разработчика, накопленные благодаря участию в аналогичных проектах.

Опыт работы с языком	Разработчик знает и умеет эффективно использовать ресурсы и ограничения языка программирования.
Навыки разработчика	Знания и навыки, которыми обладает разработчик (разрешение проблем, доступность, производительность и т.д.)
Знание темы проекта	Разработчик понимает суть проекта и знаком с концепциями, составляющими тему проекта.
Зависимость	Степени важности пользовательской истории с точки зрения ее влияния на другие истории.
Функции соединения	Это требования заказчика к обработке – входные данные, выходные данные, запросы и т.д.
Функции обработки данных	Это требования заказчика к хранению (внутренние логические файлы и внешние интерфейсные файлы).

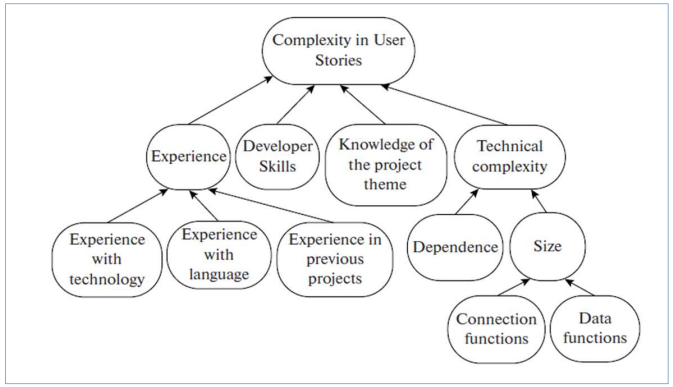

Рис. 1 Выбранная модель
Fig. 2.Selected model

6. Построение байесовской сети

6.1 Качественный компонент

Качественный компонент состоит из узлов и связей между ними, как показано на рис. 1. БС является многоуровневой иерархической структурой. Атрибуты первого уровня являются узлами, в которых собирается информация для оценки сложности; атрибуты второго уровня помогают организовать узлы иерархически; узел сложности пользовательских историй – это атрибут/узел, который показывает окончательное значение оценки сложности.

6.2 Количественный компонент

Чтобы сформировать количественный компонент, необходимо охватить три важных аспекта: значения связей между атрибутами, шкалы значений атрибутов первого уровня и построение таблицы условных вероятностей (Conditional Probability Table, CPT).

6.2.1 Значения отношений между атрибутами

Чтобы вывести значения связей между атрибутами, мы провели опрос среди профессионалов.

Опрос для подтверждения важности и значимости атрибутов модели

Цель опроса заключалась в валиации важности и значимости выбранных для модели атрибутов. Опрос проводился в цифровом формате. В опросе принял участие 21

профессионал и 19 студентов бакалавриата. При создании модели мы учитывали только ответы профессионалов. Мы подготовили вопросы для выяснения степени одобрения выбранных атрибутов, а также вопросы для сбора информации об опыте респондентов. Использовалась 5-балльная шкала Ликерта (Rensis Likert) от «полностью не согласен» до «полностью согласен».

Мы получили ответы 19 студентов, имеющих в среднем 15 месяцев опыта разработки программного обеспечения и 6 месяцев опыта использования Scrum. Мы также получили ответы от 21 профессионала, имеющих в среднем 57 месяцев опыта разработки программного обеспечения и 32 месяцев опыта использования Scrum. Обе группы использовали покер планирования для оценки пользовательских историй. Мы считаем, что участие 21 профессионала в опросе позволило нам собрать солидный объем данных в сравнении с исследованием [8], где было задействовано 16 профессионалов.

Применение уравнений

Для вычисления веса атрибутов мы использовали уравнения и определения, использованные в [8-9].

Определение 1. Пусть $V = \{v_1, \dots, v_n\}$ обозначает набор ответов профессионалов на один вопрос. Элемент $v_i = (weight, frequency)$ представляет вес одного варианта ответа по шкале Ликерта и частотность этого варианта среди всех экспертов. Формула (1) позволяет получить ненормализованное значение веса одного атрибута.

$$P_{p \in P} = \frac{\sum_{v \in V}(v_{weight} \times v_{frequency})}{\sum_{v \in V} v_{frequency}}. \quad (1)$$

Определение 2. Пусть $P = \{p_1, \dots, p_m\}$ представляет набор весов родительских узлов, которые относятся к одному и тому же дочернему узлу. Формула (2) обеспечивает нормализованные значения весов этих родительских атрибутов.

$$Pnorm_{p \in P} = \frac{P}{\sum_{p \in P} P}. \quad (2)$$

Определение 3. Пусть $A = \{a_1, \dots, a_q\}$ – это набор атрибутов второго уровня. У них имеются два состояния – *low* и *high*. Их значения вычисляются с помощью формулы (3).

$$a_{a \in A} = \frac{\sum_{p \in P} P}{|P|}. \quad (3)$$

В табл. 2 приведены частоты и веса значений атрибутов в соответствии со шкалой Ликерта. В последней строке показаны нормализованные веса атрибутов.

Табл. 2. Веса и частота значений атрибутов
Table 2. Weights and frequencies of attributes

Веса	Шкала Ликерта	Частота			
		Опыт работы с технологиями	Опыт работы с языком	Опыт работы в предыдущих проектах	Навыки разработчика
0	1	0	0	0	0
0,25	2	0	0	0	0
0,5	3	1	1	4	2
0,75	4	8	9	10	13
1	5	12	11	7	6
Вес		0,347	0,343	0,310	0,253
Веса	**Шкала Ликерта**	**Частота**			
		Знание темы проекта	**Зависимость**	**Функции соединения**	**Функции обработки данных**
0	1	0	0	0	0

0,25	2	0	1	1	1
0,5	3	3	3	3	5
0,75	4	16	12	9	6
1	5	2	5	8	9
Вес		0,234	0,490	0,504	0,496

Рассмотрим атрибуты опыта, чтобы объяснить процесс взвешивания. Применим формулу (1) для получения ненормализованного веса каждого атрибута опыта первого уровня: $Exp_with_tech = \frac{0\times0+0.25\times0+0.5\times1+0.75\times8+1\times12}{21} = 0.881$, $Exp_with_lang = 0.869$, $Exp_in_prev_proj = 0.786$. Для получения нормализованного веса воспользуемся формулой (2). Необходимо рассмотреть узлы, являющиеся родителями одного и того же дочернего узла, в данном случае: $Exp_with_tech_norm = \frac{0.881}{0.881+0.869+0.786} = 0.347$, $Exp_lang_norm = 0.343$, $Exp_prev_norm = 0.310$. Чтобы получить вес узла опыта и других узлов второго уровня, применим формулу (3): $Experience = \frac{0.881+0.869+0.786}{3} = 0.845$. Для получения нормализованного веса опыта необходимо учесть других родителей, которые влияют на один и тот же атрибут и его ненормализованный вес, в данном случае это навыки разработчика (0.798), знание темы проекта (0.738) и техническая сложность (0.770). Тогда $Experience_norm = \frac{0.845}{0.845+0.798+0.738+0.770} = 0.268$. На рис. 2 показаны результаты взвешивания узлов сети.

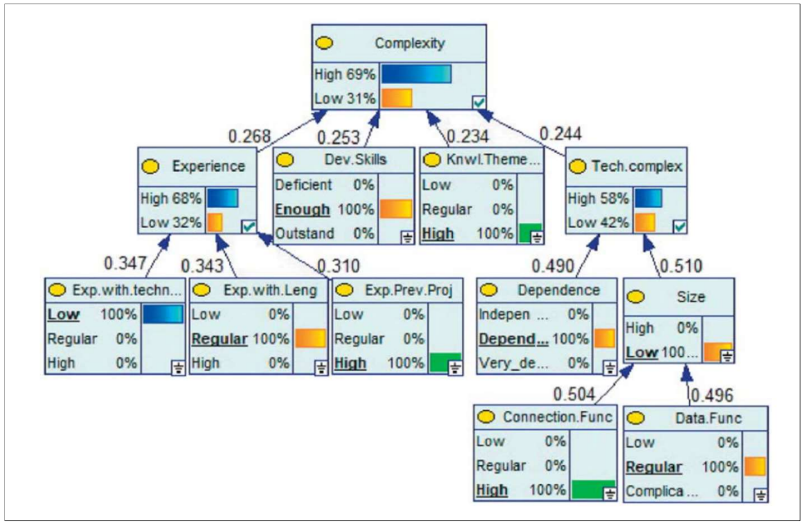

Рис. 3. Результирующие веса в байесовской сети
Fig. 4. Resulting weights in Bayesian network

Табл. 3. Вопросы для атрибутов первого уровня
Table 3. Questions for first level attributes

Опыт работы с технологиями	Как вы оцениваете свой опыт работы с технологиями, использующимися при создании пользовательской истории?	Незначительный	Я никогда не использовал технологии, с помощью которых будет разработана пользовательская история.
		Средний	Я знаком с технологиями, с помощью которых будет разработана пользовательская история, но не могу назвать себя уверенным пользователем.

		Значительный	Я уверенный пользователь технологий, с помощью которых будет разработана пользовательская история.
Опыт работы с языком	Как вы оцениваете свой опыт работы с языком программирования, использующимся при создании пользовательской истории?	Незначительный	Я никогда не имел дела с языком программирования, который будет использоваться при разработке пользовательской истории.
		Средний	Я знаком с языком программирования, который будет использоваться при разработке пользовательской истории, но не владею им уверенно.
		Значительный	Я уверенно владею языком программирования, который будет использоваться при разработке пользовательской истории.
Опыт работы в предыдущих проектах	Какой у вас опыт работы в проектах по разработке подобных пользовательских историй?	Незначительный	Я никогда не разрабатывал подобную пользовательскую историю.
		Средний	Я когда-то уже разрабатывал подобную пользовательскую историю.
		Значительный	Я разработал несколько подобных пользовательских историй.
Навыки разработчика	Как вы оцениваете свои навыки как разработчика в реализации пользовательских историй?	Недостаточные	Я не могу определить свои навыки, решение проблем дается мне с трудом.
		Достаточные	У меня есть определенные навыки, которые иногда пригождаются мне в решении проблем.
		Выдающиеся	Я хорошо владею навыками решения проблем и умею своевременно их применять.
Знание темы проекта	Как вы оцениваете свою осведомленность о теме проекта, в рамках которого разрабатывается пользовательская история?	Незначительно	Я не знаю, с чем будет связана пользовательская история
		Средне	Я знаю основы темы проекта, в рамках которого разрабатывается пользовательская история.
		Значительно	Я хорошо ориентируюсь в теме проекта, в рамках которого разрабатывается пользовательская история.
Зависимость	Насколько разрабатываемая пользовательская история зависит от других?	Не зависит	Пользовательская история не зависит от других пользовательских историй.
		Зависит	Пользовательская история немного зависит от других пользовательских историй.
		Очень зависит	Пользовательская история является основой для разработки целого ряда пользовательских историй
Функции соединения	Насколько функции соединения добавляют сложности для разработки пользовательской истории?	Незначительно	Пользовательская история содержит простые функции input, output и query.
		Средне	Пользовательская история содержит функции input, output и query средней сложности.
		Значительно	Пользовательская история содержит сложные функции input, output и query
Функции обработки данных	Насколько функции обработки данных добавляют сложности для разработки пользовательской истории?	Незначительно	Пользовательская история содержит простые функции внутренней и внешней логики.
		Средне	Пользовательская история содержит функции внутренней и внешней логики средней сложности.
		Значительно	Пользовательская история содержит сложные функции внутренней и внешней логики.

6.2.2 Априорные значения атрибутов первого уровня

Для каждого атрибута первого уровня был задан вопрос с тремя возможными ответами. В табл. 3 приведены вопросы, сгруппированные по атрибутам, и интерпретация ответов. На основе этих ответов для каждого атрибута первого уровня можно вычислить взвешенные показатели достоверности/важности количественных характеристик атрибутов. В этом исследовании мы не собирали соответствующую статистику и считали, что для всех узлов первого уровня эти показатели s одни и те же: $high\ s_1 = 0.500, regular\ s_2 = 0.330,$ и $low\ s_3 = 0.170$.

6.2.3 Составление таблиц условных вероятностей

CPT показывают вероятности событий на основе комбинаций весов атрибутов и их показателей достоверности/важности [9]. CPT составляется для каждого атрибута/узла второго уровня сети. Для получения CPT применяются определения и формулы, используемые в [8-9].

Определение 4. Пусть $S = \{s_1, s_2, \dots, s_{|S|}\}$ – набор показателей достоверности/важности. Значение каждого атрибута первого уровня умножается на значение каждого показателя достоверности/важности: $p_1 s_1, p \in P, s \in S$. Затем результаты первого шага для атрибутов первого уровня с общим узлом второго уровня объединяются: $p_1 s_1 + p_2 s_1 + \dots + p_n s_1$ и т.д. В общем виде элемент таблицы строится как

$$\sum_{1 \le i_j \le |S|, 1 \le i \le |P|} p_i\, s_{i_j}. \tag{4}$$

Для получения окончательных значений каждый элемента таблицы делится на максимальное значение элементов этой таблицы: $W = W/max$. Процесс повторяется для каждого атрибута второго уровня.

Рассмотрим атрибут второго уровня *size*, чтобы объяснить, как составлялась CPT для этого узла. Веса атрибутов первого уровня функции *connection functions* и *data functions* составили 0,504 и 0,496 соответственно (табл. 2). Для двух атрибутов CPT может быть представлена матрицей (5), для получения которой сначала необходимо умножить вес атрибута (P) на значение каждого показателя достоверности/важности (S) и рассмотреть все возможные последовательности атрибутов, образующих узел размера. Результат складывается по каждой последовательности и делится на максимальное значение матрицы для получения конечных значений:

$$\begin{bmatrix} 1.00 & 0.83 & 0.67 \\ 0.83 & 0.66 & 0.50 \\ 0.67 & 0.50 & 0.34 \end{bmatrix}. \tag{5}$$

7. Тестирование байесовской сети

7.1 Эксперименты

7.1.1 Выборка студентов

Цель этого эксперимента состояла в том, чтобы получить оценки индивидуальных пользовательских историй с помощью покера планирования и построенной сети. Была проведена выборка из 19 студентов (в 4 группах) факультета компьютерной инженерии Автономного университета Нижней Калифорнии (UABC) и 112 пользовательских историй. Студенты имели базовые знания о Scrum и покере планирования. Эксперимент состоял из 3 частей. Часть 1: Каждый член Scrum-команды оценивает каждую историю пользователя индивидуально при помощи покера планирования. Часть 2: Студенты отвечают на вопросы из табл. 3 для каждой пользовательской истории. Часть 3: Студенты оценивают истории традиционным способом. Эта часть включала повторную оценку пользовательской истории после ее выполнения.

7.1.2 Выборка профессионалов

Цель этого эксперимента состояла в том, чтобы получить оценки индивидуальных пользовательских историй с помощью покера планирования и построенной сети. Была проведена выборка из 6 профессионалов, работающих с методом Scrum и 10 пользовательских историй. Эксперимент проводился в форме цифрового опроса. Эксперимент состоял из 3 частей. Часть 1: Был выбран проект и 10 пользовательских историй, написанных студентами. Часть 2: Была предоставлена важная информация о проекте (описание проекта, язык программирования и т. д.), профессионалы должны были представить, что такая история будет разработана и оценить ее с помощью покера планирования с учетом предоставленной информации и на основе своего опыта. Часть 3: Специалистам было предложено ответить на вопросы для каждой пользовательской истории с учетом предоставленной информации.

8. Результаты и обсуждение

Краткое описание результатов эксперимента со студентами представлена в табл. 4, эксперимента с профессионалами – в табл. 5. Столбец с очками (первичная оценка) содержит результаты покера планирования до реализации пользовательской истории. Столбец со очками (финальная оценка) содержит результаты после реализации пользовательской истории. Столбец сложности представляет значение, полученное с помощью БС. Чтобы сравнить оценки в очках, выставленные с помощью покера планирования, с оценками, полученными с использованием БС, необходимо было преобразовать полученные показатели сложности в карту покера планирования, чтобы она была эквивалентна очками (SP). Были выбраны девять наиболее часто используемых карт в покере планирования: карта 1 = 1SP, карта 2 = 2SP, карта 3 = 3SP, карта 4 = 5SP, карта 5 = 8SP, карта 6 = 13SP, карта 7 = 20SP, карта 8 = 40SP и карта 9 = 100SP. Полученные показатели сложности были преобразованы в значения карт покера планирования с помощью формулы (6):

$$Y = Y_1 + \left[\left(\frac{X - X_1}{X_2 - X_1}\right)(Y_2 - Y_1)\right], \quad (6)$$

где Y – карта покера планирования, X – показатель сложности на основе БС, X1 – наименьшее значение, которое может получиться при использовании БС, X2 – наибольшее значение, которое может получиться при использовании БС, Y1 – самое низкое значение карты, Y2 – самое высокое значение карты. Столбец очков (БС) представляет эквивалентное значение в соответствии с картой, полученной после применения формулы (5). Проверка полученных результатов проводилась с помощью теста ранговой корреляции Спирмена [33]. В случае

студентов были проанализированы следующие атрибуты: *estimationIni*, *estimationFin* и *estimationBN*.

Табл. 4. Результаты эксперимента со студентами
Table 4. Students results

Член команды	Сложность	Карта (покер планирования)	Очки (первичная оценка)	Очки (финальная оценка)	Очки (БС)
Участник 1	75	4	5	5	5
Участник 2	67	3	3	5	3
Участник 3	75	4	8	5	5
Участник 4	82	5	5	5	8
Участник 5	75	4	5	5	5

Табл. 5. Результаты эксперимента с профессионалами
Table 5. Professionals results

Член команды	Сложность	Карта (покер планирования)	Очки (первичная оценка)	Очки (БС)
Участник 1	**77**	4	5	5
Участник 2	**67**	3	3	3
Участник 3	**77**	4	5	5
Участник 4	**56**	1	1	1
Участник 5	**75**	4	8	5
Участник 6	**81**	5	8	8

Тесты применялись с учетом отношений: *estimationIni-estimationBN* и *estimationFin-estimationBN*. По результатам теста Спирмена для отношения *estimationIni-estimationBN* была получена корреляция 0,659 и p-значение 0,000, а для отношения *estimationFin-estimationBN* – корреляция 0,799 и p-значение 0,000. Как видно, корреляция между финальной оценкой и оценкой, полученной с использованием байесовской сети больше, то есть оценки БС сильно приближены к оценкам студентов, уже закончивших работу над пользовательской историей.

В случае профессионалов были проанализированы переменные: *estimationIni* и *estimationBN*. Тесты применялись для отношения *estimationIni-estimationBN*. По результатам теста Спирмена для отношения *estimationIni-estimationBN* была получена корреляция 0,887 и p-значение 0,000. Если сравнить результаты студентов и профессионалов, то можно заметить, что уровень корреляции выше у вторых за счет их опыта.

В нашем исследовании предлагается модель для оценки сложности пользовательских историй. Совпадение результатов оценки, полученных с помощью предложенной модели, с результатами оценки профессиональными разработчиками – 88%. Точность результатов модели выше в случае использования опытными профессионалами, то есть модели можно доверять, как профессиональному разработчику – такой результат вполне жизнеспособен.

Если сравнить корреляцию, полученную в эксперименте с профессионалами (88%), то предложенная нами модель более точна, чем модели, предложенные в исследованиях [8] (87%) и [11] (60%). Наша модель имеет больше сходства с моделью [8], но в ней опыт разработчика разбит на три подкатегории: опыт работы с технологиями, опыт работы с языком и с предыдущими проектами. На основе этих факторов можно провести более глубокий анализ опыта как одного из наиважнейших факторов в процессе оценки. В свою очередь, в исследовании [10] точность оценок достигает 90%, но сравнивать нашу модель с этим результатом нельзя, поскольку там декомпозиция сложности производилась с учетом технических подкатегорий. Несмотря на различный характер атрибутов, достигнутая нами точность была приближена к результатам [10].

9. Заключение

В настоящем исследовании был предложен метод для оценки пользовательских историй с помощью модели, состоящей из атрибутов с упором на индивидуальные качества разработчиков. С помощью этой модели мы стремились повысить точность оценки пользовательских историй. Мы взяли за основу ранее опубликованную нами статью [28] и проанализировали упомянутые в ней атрибуты, чтобы сформировать предложенную модель. С помощью анализа были отобраны наиболее релевантные из них: опыт работы с технологиями, опыт работы с языком и опыт работы в предыдущих проектах, навыки разработчика, знание темы проекта. Мы добавили несколько важных атрибутов, связанных с технической сложностью пользовательских историй: зависимость и размер.

Для тестирования нашей модели мы использовали две выборки респондентов – студентов и профессионалов. Для проверки полученных результатов мы применили корреляционные тесты к полученным оценкам. В случае со студентами результаты показывают, что данные, полученные с помощью сети, имеют схожесть с оценками, сформулированными после выполнения пользовательских историй. В эксперименте с профессионалами показатели корреляции выше, чем у студентов, что связано с опытностью первых. В результате оценки, полученные с помощью нашей модели, на 88% совпадают с оценками профессиональных разработчиков, что говорит об успешности и достоверности исследования.

Сравнение результатов нашего исследования с похожими работами – исследованиями [8] с аналогичной методологией и [11] с методологией, отличной от нашей, но схожим подходом – показало, что наша модель дает более точные результаты, возможно, за счет того, что оно основано на индивидуальных возможностях разработчиков, имеющих большое влияние на процесс оценки. В нашей модели мы рассмотрели опыт, но не как единый фактор. Разбив на составляющие такой важный атрибут, как опыт, мы смогли повысить точность наших оценок.

Результаты, полученные с помощью байесовской сети, обладают высокой достоверностью, поскольку они схожи с результатами оценки профессиональными разработчиками; успех обусловлен тем, что вес атрибутов в сети основан на мнении профессионалов, это помогает справляться с ситуацией неопределенности и неопытности разработчиков и повысить точность оценки. Метод будет особенно полезен в работе с неопытными разработчиками и поможет обрести уверенность в процессе оценки. В будущем планируется расширение модели для повышения точности оценок, а также увеличение выборки команд для получения еще более проверенных данных.

Список литературы / References

[1] K.S. and J. Sutherland. The Scrum Guide. The Definitive Guide to Scrum: the Rules of the Game. Scrum.Org and Scrum Inc, 2017, 19 p.

[2] A. Srivastava, S. Bhardwaj, and S. Saraswat. SCRUM model for agile methodology. In Proc. of the International Conference on Computing, Communication and Automation, 2017, pp. 864-869.

[3] K.S. Rubin. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professiona, 2012, 496 p.

[4] D. Pauly, B. Michalik, and D. Basten. Do Daily Scrums Have to Take Place Each Day? A Case Study of Customized Scrum Principles at an E-commerce Company. In Proc. of the 48th Hawaii International Conference on System Sciences, Kauai, 2015, pp. 5074-5083.

[5] A Guide to the Scrum Body of Knowledge (SBOK Guide). VMEdu, Inc, 2016. 140 p.

[6] G. Lucassen, F. Dalpiaz, J.M E.M. van der Werf, and S. Brinkkemper. Improving agile requirements: the Quality User Story framework and tool. Requirements Engineering, vol. 21, no. 3, 2016, pp. 383-403.

[7] J. Lopez-Martinez, A. Ramirez-Noriega, R. Juarez-Ramirez et al. Analysis of Planning Poker Factors between University and Enterprise. In Proc. of the 5th International Conference in Software Engineering Research and Innovation, 2017, pp. 54–60.

[8] J. López-Martínez, A. Ramírez-Noriega, R. Juárez-Ramírez et al. User stories complexity estimation using Bayesian networks for inexperienced developers. Cluster Computing, vol. 21, no. 2, 2018, pp. 715-728.

[9] J. López-Martínez, R. Juárez-Ramírez, A. Ramírez-Noriega et al. Estimating User Stories' Complexity and Importance in Scrum with Bayesian Networks. Advances in Intelligent Systems and Computing, vol. 569, 2017, 205-214.

[10] S. Dragicevic, S. Celar, and M. Turic. Bayesian network model for task effort estimation in agile software development. Journal of Systems and Software, vol. 127, 2017, pp. 109-119.

[11] J.M. Alostad, L.R.A. Abdullah, and L.S. Aali. A Fuzzy based Model for Effort Estimation in Scrum Projects. International Journal of Advanced Computer Science and Applications, vol. 8, no. 9, 2017, pp. 270-277.

[12] E. Scott and D. Pfahl. Using developers' features to estimate story points. In Proc. of the 2018 International Conference on Software and System Process, 2018, pp. 106-110.

[13] V. Lenarduzzi. Could social factors influence the effort software estimation? In Proc. of the 7th International Workshop on Social Software Engineering, 2015, pp. 21-24.

[14] V. Mahnič and T. Hovelja. On using planning poker for estimating user stories. Journal of Systems and Software, vol. 85, no. 9, 2012, pp. 2086-2095.

[15] S.Z. Ziauddin, S.K. Tipu, and S. Zia. An Effort Estimation Model for Agile Software Development. Advances in Computer Science and Its Applications, vol. 2, no. 1, 2012, pp. 2166-2924.

[16] B. Peischl, M. Zanker, M. Nica, and W. Schmid. Constraint-based recommendation for software project effort estimation. Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 4, 2010, pp. 282-290.

[17] F. Zare, H. Khademi Zare, and M.S. Fallahnezhad. Software effort estimation based on the optimal Bayesian belief network. Applied Soft Computing, vol. 49, 2016, pp. 968–980.

[18] S.K. Rath, B.P. Acharya, and S.M. Satapathy. Early stage software effort estimation using random forest technique based on use case points. IET Software, vol. 10, no. 1, 2016, pp. 10-17.

[19] S. C. Hsu, K.W. Weng, Q. Cui, and W. Rand. Understanding the complexity of project team member selection through agent-based modeling. International Journal of Project Management, vol. 34, no. 1, 2016, pp. 82-93.

[20] O. Malgonde and K. Chari. An ensemble-based model for predicting agile software development effort. Empirical Software Engineering, vol. 24, issue 2, 2018, pp. 1017-1055.

[21] R. Sholiq, S. Dewi, and A.P. Subriadi. A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points. Procedia Computer Science, vol. 124, 2017, pp. 470-477.

[22] A. Qazi, J. Quigley, A. Dickson, and K. Kirytopoulos. Project Complexity and Risk Management (ProCRiM): Towards modelling project complexity driven risk paths in construction projects International Journal of Project Management, vol. 34, no. 7, 2016, pp. 1183-1198.

[23] C. Drahý and O. Pastor. Relationship Between Project Complexity and Risk Kinds Identified on Projects. EMI – Economics, Management, Innovation, vol. 8, no. 1, 2016, pp. 42-50.

[24] S.M. Abdou, K. Yong, and M. Othman. Project Complexity Influence on Project management performance – The Malaysian perspective. In Proc. of the 4th International Building Control Conference, 2016, article no. 00065.

[25] H. Zahraoui and M.A. Janati Idrissi. Adjusting story points calculation in scrum effort & time estimation. In Proc. of the 10th International Conference on Intelligent Systems: Theories and Applications, 2015, pp. 1-8.

[26] E. Mendes. Knowledge Representation using Bayesian Networks A Case Study in Web Effort Estimation. In Proc. of the World Congress on Information and Communication Technologies, 2011, pp. 612-617.

[27] Р. Сильхавы, З. Попова, П. Сильхавы. Алгоритмический метод оптимизации оценки трудозатрат. Программирование, том 42, no. 3, 2016 г., стр. 64-71 / R. Silhavy, Z. Prokopova, and P. Silhavy. Algorithmic optimization method for effort estimation. Programming and Computer Software, vol. 42, no. 3, 2016, pp. 161-166.

[28] M. Durán, R. Juárez-Ramírez, S. Jiménez, and C. Tona. Taxonomy for Complexity Estimation in Agile Methodologies: A Systematic Literature Review. In Proc. of the 7th International Conference in Software Engineering Research and Innovation, 2020, pp. 87-96.

[29] M. Usman, E. Mendes, F. Weidt, and R. Britto. Effort estimation in Agile Software Development: A systematic literature review. In Proc. of the 10th International Conference on Predictive Models in Software Engineering, pp. 82–91, 2014.

[30] A. Georgsson. Introducing Story Points and User Stories to Perform Estimations in a Software Development Organisation – A case study at Swedbank IT. Master's Thesis, UMEÅ University, 2011, 52 p.

[31] N. R. Jennings and M. Wooldridge. Agent-Oriented Software Engineering. Artificial Intelligence, vol. 117, 2000, pp. 277-296.

[32] F.V. Jensen. Bayesian networks. WIREs Computational Statistics, vol. 1, no. 3, 2009, pp. 307–315.
[33] L. Myers, M.J. Sirois. Spearman Correlation Coefficients, Differences between. In Encyclopedia of Statistical Sciences. John Wiley & Sons, 2006.

Информация об авторах / Information about authors

Майра ДУРАН – магистр, профессор. Область научных интересов: гибкие методологии, оценка программных проектов.

Mayra DURÁN, Master of Science, Professor. Research interests include agile methodologies, estimation of software projects.

Рейес ХУАРЕС-РАМИРЕС, кандидат компьютерных наук, профессор. Область научных интересов: программная инженерия, оценка неопределенности программного обеспечения и взаимодействие человека с компьютером.

Reyes JUÁREZ-RAMÍREZ, Doctor of Computer Science, Full Professor. Research interests include software Engineering, software uncertainty estimation, and human-computer interaction.

Саманта ХИМЕНЕС, доктор наук, профессор. Область научных интересов: программная инженерия, удобство использования, образовательные технологии, взаимодействие человека и компьютера.

Samantha JIMÉNEZ, Doctor of Science, Full Professor. Research interests include Software Engineering, Usability, Educational Technology, Human-Computer Interaction.

Клаудия ТОНА, магистр, профессор. Сфера научных интересов: программная инженерия, гибкие методологии, Scrum, гибкие фреймворки.

Claudia TONA, Master of Science, Professor. Research interests include Software Engineering, Agile Methodologies, Scrum, Agile Frameworks.