

МОДЕЛЬ ОЦЕНКИ КВАЛИФИКАЦИИ ИТ-СПЕЦИАЛИСТОВ В ОБЛАСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ*

Э.И. ГАВРИЛЬЕВ¹, Т.В. АВДЕЕНКО²

¹ 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры теоретической и прикладной информатики. E-mail: erchimen_gavriliev@outlook.com

² 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, профессор кафедры теоретической и прикладной информатики. E-mail: tavdeenko@mail.ru

Решающим фактором успешной работы современных ИТ-организаций являются профессиональные качества персонала и система их развития. Развитие персонала для ИТ-организаций в сфере разработки программного обеспечения (ПО) имеет особое значение в связи с тем, что деятельность предъявляет высокие требования к уровню квалификации сотрудников, их знаниям и навыкам, тогда как они имеют свойство быстро терять свою актуальность. Однако при оценке квалификации персонала возникают проблемы, влияющие на объективность оценки и эффективность дальнейших управленческих решений, например: фаворитизм, предвзятость, высокая требовательность, ошибочный эталон. Соответственно, целью работы является разработка модели оценки квалификации ИТ-специалистов в области разработки программного обеспечения. В работе рассматриваются система оценки персонала, виды критериев оценки квалификации разработчиков и тестировщиков программного обеспечения, которые охватывают технические аспекты деятельности, межличностные и ментальные характеристики специалистов. Для оценки квалификации была выбрана и изменена модель концептуальной теории компетентности разработки ПО. На основе обзора литературы, анализа вакансий и интервью с ИТ-специалистами были выделены основные направления деятельности разработчиков и тестировщиков, а также характеристики, которые необходимо оценить. На основе измененной модели был разработан прототип системы на языке программирования JavaScript и на программной платформе Node JS сбора информации, который загружает данные из систем контроля версий и отслеживания задач Atlassian Jira для дальнейшей оценки.

Ключевые слова: разработчики программного обеспечения, тестировщики программного обеспечения, компетенции, методы оценки персонала, критерии оценки персонала, управление персоналом, управление карьерой, теория процессов, NodeJS

* Статья получена 19 апреля 2021 г.

ВВЕДЕНИЕ

В настоящее время динамично меняются факторы внутренней и внешней среды ИТ-индустрии: цифровая трансформация множества отраслей бизнеса, высокая сложность разрабатываемых систем из-за специфичных требований заказчиков, большая скорость появления новых технологий.

В рамках данных условий ИТ-организациям необходимо ориентировать свою деятельность на повышение конкурентного потенциала. Одним из основных факторов конкурентоспособности организации являются персонал и система его развития.

Однако при развитии профессиональных навыков ИТ-специалистов возникают следующие проблемы:

- 1) оценивание персонала – трудоемкий процесс, требующий значительных временных ресурсов, результаты которого являются субъективными;
- 2) существующие критерии и методы оценки фокусируют свое внимание только на технические навыки;
- 3) возникают трудности при подборе пути развития профессиональных навыков сотрудников.

Целью настоящей работы является разработка модели оценки квалификации ИТ-специалистов в области разработки программного обеспечения.

1. ОБЗОР АКТУАЛЬНЫХ РАБОТ

1.1. СИСТЕМА ОЦЕНИВАНИЯ ПЕРСОНАЛА

Оценивание персонала тесно связывают с системным подходом, система оценки персонала представляет собой совокупность взаимосвязанных действий, которые направлены на сбор данных о соответствии сотрудников заданным критериям [1].

В состав элементов системы оценивания персонала входят [2]:

- субъект оценки;
- объект оценки;
- предмет оценки;
- метод оценки;
- критерии оценки.

Объектом оценки является сотрудник либо группа сотрудников, которая выделена по определенному признаку (например, по занимаемой должности).

Субъектом оценки чаще всего является непосредственный руководитель, который составляет характеристику оцениваемому объекту. Также субъектом оценки могут выступать HR-отдел, коллеги, которые имеют функциональную связь с объектом оценки, и независимые эксперты.

Предмет оценки – характеристики объекта оценки (например, личностные и профессиональные качества, результаты работы и т. д.).

Критерии оценки – показатели и характеристики, позволяющие определить эффективность выполнения своих обязанностей непосредственным исполнителем [3].

В отечественной и мировой практике используется большое число методов оценки персонала. Выделяют три группы методов оценивания персонала [4]:

- 1) качественные – методы, которые предоставляют характеристики сотрудников без их количественного выражения;
- 2) количественные – методы, при помощи которых можно получить числовую оценку оцениваемых качеств сотрудников;
- 3) комбинированные – объединяют в себе особенности и приемы как качественных, так и количественных методов.

1.2. КРИТЕРИИ ДЛЯ ОЦЕНИВАНИЯ ДЕЯТЕЛЬНОСТИ РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Выделить объективные критерии для оценки разработчиков ПО довольно затруднительно. Большое количество работ акцентирует свое внимание только на технические навыки. В работе Суракка проведено исследование, в котором разработчики ПО оценили важность 42 различных предметов и навыков, ориентированных в большей мере на навыки, связанные с программированием: объектно-ориентированное программирование, функциональное программирование, интернет-протоколы, структуры данных и т. д. [5]. Исследование Робилларда и других авторов посвящено оценке эффективности анализа исходного кода системы, так как при разработке, поддержке и сопровождении часто приходится вносить изменения в уже разработанную систему [6]. Работа Ахмадзаде фокусирует свое внимание на оценке эффективности поиска и устранения дефектов в исходном коде системы [7].

Другая часть исследований сосредоточена на человеческих и социальных факторах, но она не связывает эти факторы с техническими навыками. Например, в работе Фахима проведен анализ 500 вакансий в области ИТ из разных стран, в котором основной фокус был уделен таким софт-скиллам, как коммуникационные навыки, организационные навыки, адаптивность и т. д. [8].

Некоторые работы предоставляют более целостное видение навыков разработчиков, которое включает в себя технические, межличностные и ментальные атрибуты, но они не совсем полны в своих исследованиях.

Например, исследование Хьюнера и Гуздиала посвящено компетенциям, которые работодатели из игровой индустрии ищут в выпускниках университетов [9]. В результате наиболее важными были выделены такие навыки, как программирование, а также навыки работы с людьми (например, «способность работать с другими» и «сдерживать свое эго»).

В статье Бегеля и Саймона было проведено похожее исследование, в котором проводился анализ работы восьми новых сотрудников Microsoft в течение четырех недель [10]. Авторы выявили, что новички должны быть «настойчивыми» и эффективно сотрудничать в «большой команде разработчиков ПО».

Описанные выше работы концентрируют свое внимание на «новичках» в ИТ-области, что не совсем подходит для оценки опытных специалистов.

Наиболее актуальными работами являются исследование Ли и Ко, посвященное поиску и приоритизации атрибутов «отличных» разработчиков ПО (great software engineer), и исследование Балтеса и Дилия, в котором представлена концептуальная теория компетентности разработки ПО.

В работе Ли и Ко был проведен опрос 1926 сотрудников Microsoft о характеристиках «отличного» разработчика ПО [11]. На основе результатов опроса было выделено 54 атрибута, которые можно разделить на 4 группы (рис. 1) [12].

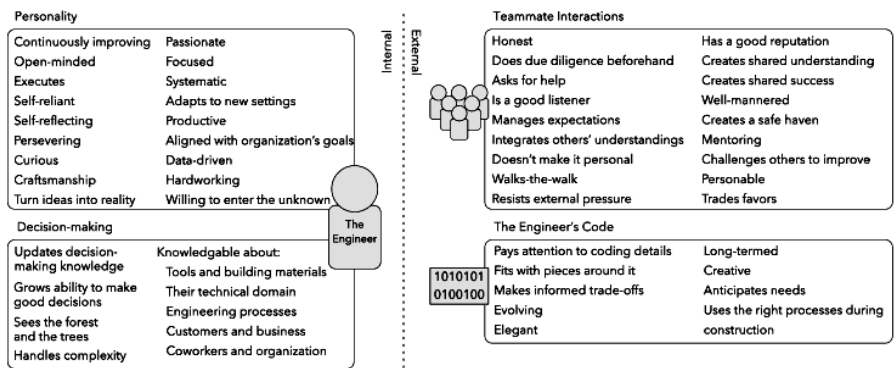


Рис. 1. Модель атрибутов «отличного» разработчика ПО

Fig. 1. Model of attributes of great software engineers

1. Группа «Личные характеристики» (Personality) включает 18 атрибутов, описывающих личность разработчиков ПО. Примерами атрибутов являются увлеченность (Passionate) и любопытство (Curious).
2. Группа «Принятие решений» (Decision-making) включает 9 атрибутов, связанных со способностью разработчиков принимать решения. Под процессом принятия решений исследователи понимают оценку текущей ситуации, определение альтернативных вариантов действий, оценку вероятностных результатов и оценку ценности будущих состояний.

3. Группа «Работа с командой» (Teammates) состоит из 17 атрибутов, описывающих взаимодействие разработчиков с коллегами. Атрибуты в этой области основаны на четырех концепциях: быть разумным человеком, быть хорошим лидером, эффективно общаться и укреплять доверие.

4. Группа «Программный продукт» (Software Product) состоит из девяти атрибутов программного обеспечения, созданного разработчиками, например, элегантный (Elegant) и креативный (Creative).

Самый высокий рейтинг важности получили следующие атрибуты:

- обращает внимание на детали программирования (pays attention to coding details);
- способен справляться со сложностями (mentally capable of handling complexity);
- постоянно совершенствуется (continuously improving);
- честный (honest).

Самый низкий рейтинг важности получили такие атрибуты:

- услужливый (trades favors);
- трудолюбивый (hardworking).

В работе Балтеса и Дили разработана концептуальная теория компетентности разработки программного обеспечения (SDExp) [13]. Теория основана на ответах онлайн-опроса 355 разработчиков ПО и на существующей литературе в области психологии по экспертным знаниям и производительности. Модель теории представлена на рис. 2.

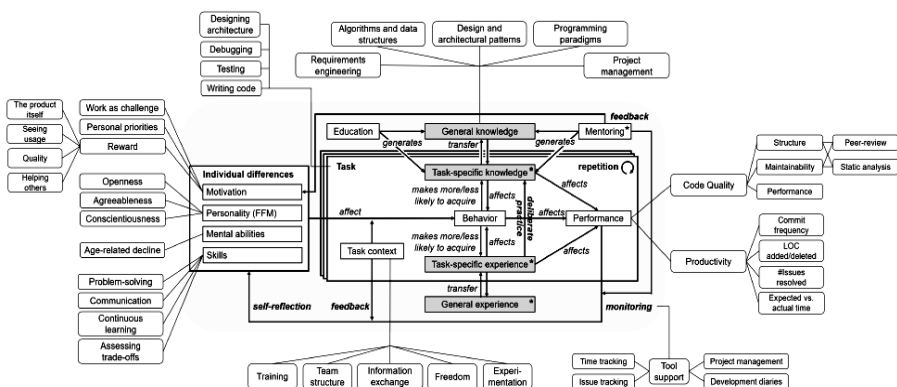


Рис. 2. Модель концептуальной теории компетентности разработки ПО

Fig. 2. Model of conceptual theory of software development expertise

Модель теории включает в себя элементы двух видов – это конструкты и взаимосвязи между этими конструктами. В состав конструктов входит следующее.

Индивидуальные различия (Individual differences). В данном конструкте указаны конкретные характеристики экспертов, в частности черты характера, которые способствуют развитию экспертных знаний. Например, непредвзятость (open-minded), любопытность (curious), способность к работе в команде (team players). К данной группе конструктов также отнесли такие конструкты, как мотивация (motivation), черты характера (personality), когнитивные возможности (mental abilities) и общие навыки (skills).

Задачи (Tasks). Этот конструкт включает в себя 3 наиболее важные задачи, решаемые разработчиком: разработка архитектуры ПО (designing software architecture), написание исходного кода (writing source code) и анализ и интерпретация требований (analyzing and understanding requirements).

Опыт, знания (Experience, knowledge). К классу этих конструктов включены общие характеристики экспертов по разработке ПО: общий опыт (general experience), опыт работы над задачей (task-specific experience), общие знания (general knowledge) и «глубокие» знания (task-specific knowledge). К примеру, к общим знаниям относят знания о парадигмах программирования, структурах данных, алгоритмах и подходах к проектированию.

Наставничество (mentoring). К этому конструкту относят характеристики ментора, так как более половины участников опроса работали с наставником, который помогал им.

Производительность (performance). В этот конструкт включены метрики, при помощи которых можно измерить производительность экспертов.

В состав взаимосвязей включены **мониторинг, обратная связь и самоанализ (Monitoring, feedback and self-reflection)**. Наиболее важной деятельностью по мониторингу была экспертная оценка (peer review). Респонденты упоминали, что просили коллег дать обратную связь, выполняли анализ кода или участвовали в парном программировании. Участники также упомянули инструменты отслеживания времени (time tracking) и отслеживания ошибок (issue tracking) в качестве источников обратной связи, сравнивая ожидаемые с фактическими результатами.

1.3. КРИТЕРИИ ДЛЯ ОЦЕНКИ ТЕСТИРОВЩИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Тестирование программного обеспечения является неотъемлемой частью разработки ПО, которое обеспечивает качество продуктов и удобство работы с ними. Несмотря на то что тестирование требует определенных знаний и

навыков, углубленные исследования деятельности тестировщиков несколько отстают от исследований других ролей в командах по разработке ПО.

CSTE СВОК (Certified Software Tester Common Body of Knowledge) представляет собой ядро соответствующих фактических знаний в области тестирования ПО [14]. Его можно использовать в качестве руководства для разработки должностных инструкций ПО, а также для оценки компетенций тестировщиков.

В CSTE СВОК навыки тестирования были разделены на 10 категорий.

1. Принципы и концепции тестирования ПО (Software Testing Principles and Concepts).
2. Сборка тестовой среды (Building the Test Environment).
3. Управление тестовым проектом (Managing the Test Project).
4. Планирование тестирования (Test Planning).
5. Выполнение плана тестирования (Executing the Test Plan).
6. Процесс создания отчетов об испытаниях (Test Reporting Process).
7. Приемочное тестирование пользователей (User Acceptance Testing).
8. Программное обеспечение для тестирования, разработанное подрядчиками (Testing Software Developed by Contractors).
9. Тестирование средств управления программным обеспечением и адекватность процедур безопасности (Testing Internal Control).
10. Тестирование новых технологий (Testing New Technologies).

Для сертификации тестировщиков авторами была предложена трехуровневая шкала владения описанными выше навыками.

1. Не компетентен (Not competent) – эта оценка означает, что тестировщик не понимает этот навык или не знает, что требуется для выполнения этого навыка.

2. Некоторый уровень компетентности (Some competency) – эта оценка означает, что тестировщик знает, что необходимо для достижения определенного навыка.

3. Полностью компетентен (Fully competent) – эта оценка означает, что тестировщик не только знает, что требуется для выполнения определенного навыка, но и фактически применял этот навык в повседневной работе.

В работе Йивонен и Мантила на основе интервью со специалистами из ИТ-области были изучены характеристики эффективных тестировщиков программного обеспечения [15]. В результате были выделены 4 класса характеристик:

- 1) опыт и навыки (Experience and skills);
- 2) анализ (Reflection);
- 3) мотивы (Motivation);
- 4) личные характеристики (Personal Characteristics).

Среди найденных характеристик знания предметной области и специализированные технические навыки считаются более важными, чем навыки в разработке тест-кейсов и планировании тестирования.

Исследование Дика посвящено поиску и анализу наиболее важных характеристик для тестировщика [16]. В этом исследовании было проведено несколько интервью с тестировщиками и менеджерами тестирования из четырех ИТ-компаний. В состав важных характеристик тестировщиков были включены навыки коммуникации (Communication), образование в сфере информатики (IT Background), внимание к деталям (Details oriented), любопытство (Curious) и терпеливость (Patient).

Наиболее актуальной является работа Флорея и Стрэй, в которой был проведен анализ 400 вакансий в 33 странах для разработки профиля навыков тестировщиков, необходимых отрасли в настоящее время [17]. Исследователи выделили 4 группы навыков:

- 1) навыки, связанные с тестированием (Software Testing skills);
- 2) технические навыки (Technical skills);
- 3) софт навыки (Soft skills);
- 4) навыки, связанные с предметной областью (Domain-specific skills).

Результаты показывают, что работодателей больше всего интересуют навыки, связанные с планированием и разработкой тестов, автоматизацией тестирования, функциональным тестированием, тестированием производительности и отчетами о ходе работы. Тестировщики должны обладать техническими навыками (например, языками программирования). Кроме того, они должны выполнять задачи, связанные с управлением проектами, такие как оценка, управление рисками и обеспечение качества.

2. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

2.1. МОДЕЛЬ ОЦЕНКИ КВАЛИФИКАЦИИ ИТ-СПЕЦИАЛИСТОВ

Разработку ПО чаще всего декомпозируют на набор задач, например, анализ и сбор требований, программирование, тестирование и т. д. В связи с этим особую актуальность приобретает концептуальная теория компетентности разработки программного обеспечения Балтеса и Дилия, ориентированная на задачи.

Для оценки квалификации ИТ-специалистов в области разработки программного обеспечения предлагается использовать модель, схема которой представлена на рис. 3.

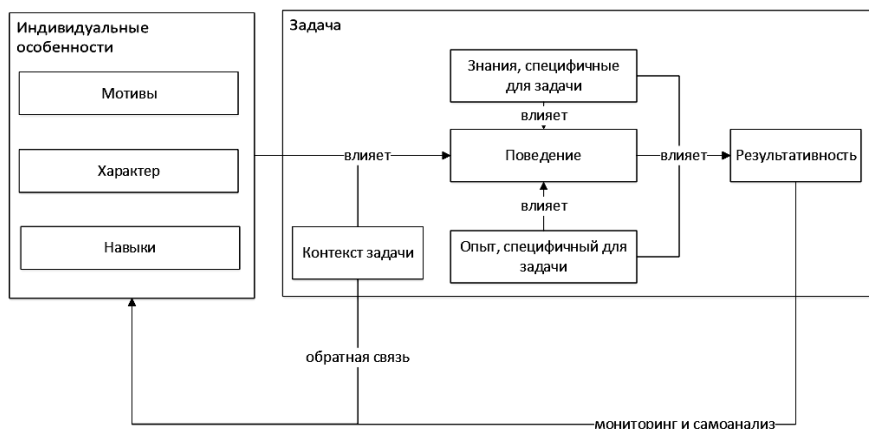


Рис. 3. Модель оценки квалификации ИТ-специалистов

Fig. 3. Model for the assessment of IT-specialists' qualification

Уровень соответствия занимаемой должности определяется результативностью, показатели которой зависят от выполняемой задачи. К конструкту «результативность» относятся метрики, при помощи которых можно измерить производительность экспертов. К конструкту «задачи» относят основные задачи, которые выполняет ИТ-специалист в своей ежедневной работе.

При решении задачи специалист использует знания и опыт, специфичные для этой задачи, также на поведение сотрудника влияет контекст задачи – организационные, социальные и технические ограничения, которые имеют отношение к поставленной задаче.

Сам специалист характеризуется набором показателей, составляющих 3 группы индивидуальных особенностей: мотивы, характер и навыки, которые, в свою очередь, влияют на поведение сотрудника.

На основе обзора литературы, вакансий в области ИТ, должностных инструкций и комментариев, собранных в ходе интервью с ИТ-специалистами, были определены основные задачи для разработчиков и тестировщиков. Основными задачами разработчиков являются:

- 1) анализ требований;
- 2) написание исходного кода;
- 3) разработка архитектуры ПО;
- 4) отладка ПО.

Для оценки квалификации сотрудника при выполнении задачи «Отладка ПО» выделим следующие элементы конструктов модели (табл. 1).

Т а б л и ц а 1

Table 1

Элементы конструкторов для оценки квалификации сотрудника при выполнении задачи «Отладка ПО»

Elements of model's constructs for the assessment of employee's expertise in execution of the task "Debugging"

| Название конструктора | Элемент |
|--------------------------------|---|
| Знания, специфичные для задачи | Знания языка программирования |
| | Знания инструментов для разработки |
| | Знания предметной области разрабатываемой системы |
| | Знания систем контроля версий |
| Опыт, специфичный для задачи | Количество исправленных ошибок до проведения оценки |
| Контекст задачи | Язык программирования |
| | Архитектура системы |
| | Шаги для воспроизведения ошибки |
| Результативность | Количество исправленных ошибок |
| | Средняя длительность исправления ошибок |
| | Среднее количество переоткрытий задач по исправлению ошибок |

В список основных задач тестировщиков включено:

- 1) написание тест-кейсов;
- 2) планирование тестирования;
- 3) проведение ручного тестирования;
- 4) разработка автотестов.

Для оценки квалификации сотрудника при выполнении задачи «Проведение ручного тестирования» выделим следующие элементы конструкторов модели (табл. 2).

Таблица 2

Table 2

Элементы конструкторов для оценки квалификации сотрудника при выполнении задачи «Проведение ручного тестирования»

Elements of model's constructs for the assessment of employee's expertise in execution of the task "Manual testing"

| Название конструкта | Элемент |
|--------------------------------|---|
| Знания, специфичные для задачи | Знания в области тестирования |
| | Знания инструментов для управления процессом тестирования |
| | Знания в области видов тестирования |
| | Знания в области стратегий тестирования |
| Опыт, специфичный для задачи | Количество проверенных задач до проведения оценки |
| | Длительность работы с системой |
| Контекст задачи | Язык программирования |
| | Архитектура системы |
| Результативность | Количество проверенных задач |
| | Среднее время проверки задачи |

2.2.. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРОТОТИПА СИСТЕМЫ СБОРА ИНФОРМАЦИИ

Чаще всего для оценки кандидатов используют следующие основные источники информации:

- документы из личного дела;
- нормативные документы, регламентирующие деятельность работника;
- мнения начальника, коллег, подчиненных.

Однако данные из этих источников слабо поддаются формализации, в связи с чем особую актуальность приобретают такие источники, как системы отслеживания задач (Task tracking system / Bug tracking system) и системы контроля версий (Version Control System).

Взаимодействие в рамках разработки программного обеспечения (регистрация задач и ошибок, мониторинг решения задач и исправления ошибок и т. д.) осуществляется с помощью системы отслеживания ошибок, предназначенной для организации взаимодействия проектной команды Atlassian Jira [18]. В Jira каждый запрос представляет собой задачу, которую необходимо выполнить в рамках разработки или сопровождения информационной системы [19].

Основными полями, описывающими запрос, являются:

- 1) приоритет (Priority) – важность запроса;
- 2) статус (Status) – текущее состояние запроса;
- 3) исполнитель (Assignee) – сотрудник, ответственный за решение запроса;
- 4) автор (Reporter) – сотрудник, создавший запрос;
- 5) журнал работ (Worklogs) – записи, хранящие информацию о количестве времени, уделенного на решение запроса;
- 6) история изменений (Changelogs) – записи, хранящие информацию об изменениях запроса.

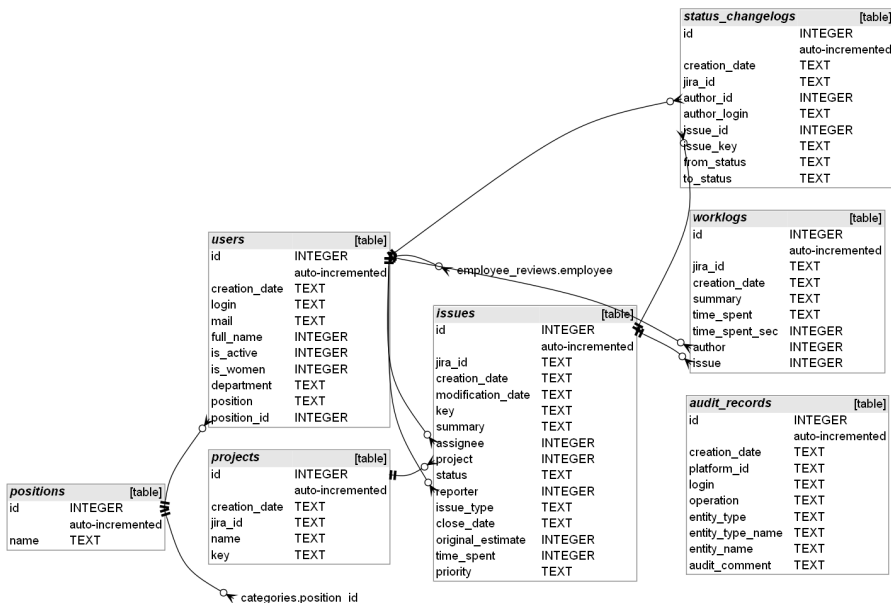


Рис. 4. Схема данных прототипа информационной системы

Fig. 4. Database schema of the prototype

На основе полей, описанных выше, можно получить количественные данные о сроках выполнения задачи, количестве затраченного времени на исполнение и количестве переоткрытий задачи.

Система контроля версий (СКВ) – это система, которая фиксирует изменения кода программного обеспечения, внесенные разработчиками [20]. Наиболее распространенными системами контроля версий являются Git, SVN, Mercurial и др.

На основе данных из СКВ можно получить более подробную информацию о количестве измененных разработчиком строк кода, частоте изменения кода информационной системы, а также сам код, который написал разработчик.

Для сбора информации из этих систем был разработан прототип на языке программирования JavaScript на программной платформе Node.js, в качестве системы управления базой данных была использована SQLite. Схема данных прототипа представлена на рис. 4.

На основе собранных данных из систем отслеживания задач и контроля версий в дальнейшем можно провести оценку результативности сотрудника по различным видам задач. Например, на основе данных из истории изменения запроса в Jira можно определить среднее время решения задачи, среднее время проверки решения задачи и количество переоткрытий.

ЗАКЛЮЧЕНИЕ

В рамках настоящей работы для оценки квалификации была выбрана и изменена модель концептуальной теории компетентности разработки ПО. На основе обзора литературы, анализа вакансий и интервью с ИТ-специалистами были выделены основные направления деятельности и характеристики, которые необходимо оценить. На основе измененной модели был разработан прототип системы сбора информации, который загружает данные из систем контроля версий и отслеживания задач для дальнейшей оценки. В дальнейшем планируется выделить критерии для бизнес-аналитиков и ведущих разработчиков и доработать прототип для проведения оценки квалификации ИТ-специалистов.

СПИСОК ЛИТЕРАТУРЫ

1. Носырева И.Г. Анализ эффективности системы оценки персонала // Экономика труда. – 2019. – № 1 (6). – С. 440–452.
2. Кибанов А. Я. Управление персоналом в России: теория, отечественная и зарубежная практика. – М.: ИНФРА-М, 2014. – 283 с.

3. Джинчарадзе Г.Р. Процедура оценки персонала в рамках системы управления человеческими ресурсами // Наукоедение. – 2013. – № 3. – С. 1–12.
4. Гафурова О.Ю. Качественные, количественные и комбинированные методы оценки персонала // Актуальные проблемы гуманитарных и естественных наук. – 2017. – № 5 (4). – С. 55–58.
5. Surakka S. What subjects and skills are important for software developers? // Communications of the ACM. – 2007. – Vol. 50 (1). – P. 73–80.
6. Robillard M.P., Coelho W., Murphy G.C. How effective developers investigate source code: an exploratory study // IEEE Transactions on Software Engineering. – 2004. – Vol. 30 (12). – P. 889–903.
7. Ahmadzadeh M. An analysis of patterns of debugging among novice computer science students // ACM SIGCSE Bulletin. – 2005. – Vol. 37 (3). – P. 84–88.
8. Faheem A. Soft skills requirements in software development jobs: a cross cultural empirical study / F. Ahmed, L. Fernando Capretz, S. Bouktif, P. Campbell // Journal of Systems and Information Technology. – 2005. – Vol. 14 (1). – P. 58–81.
9. Hewner M., Guzdial M. What game developers look for in a new graduate: interviews and surveys at one game company // Proceedings of the 41st ACM Technical Symposium on Computer Science Education. – New York, 2010. – P. 275–279.
10. Begel A., Simon B. Novice software developers, all over again // Computing Education Research Workshop. – Sydney, Australia, 2008. – P. 3–14.
11. Li P.L., Ko A.J., Begel A. What distinguishes great software engineers? // Empirical Software Engineering. – 2020. – Vol. 25. – P. 322–352.
12. Li P.L., Ko A.J., Zhu J. What makes a great software engineer? // 37th IEEE International Conference on Software Engineering. – 2015. – Vol. 1. – P. 700–711.
13. Baltes S., Diehl S. Towards a theory of software development expertise // Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018. ACM. – Lake Buena Vista, FL, 2018. – P. 187–200.
14. Certified Software Tester (CSTE): web-сайт. – URL: <https://www.softwrecertifications.org/cste/> (accessed: 30.05.2021).
15. Iivonen J., Mäntylä M., Ikonen J. Characteristics of high performing testers: a case study // ESEM' 10: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. – Bolzano-Bozen, Italy, 2010. – P. 1–9.
16. Deak A. What characterizes a good software tester? – A survey in Four Norwegian Companies // Testing Software and Systems. ICTSS 2014. – Berlin;

Heidelberg: Springer, 2014. – P. 161–172. – DOI: 10.1007/978-3-662-44857-1_11. – (Lecture Notes in Computer Science; vol. 8763).

17. *Florea R., Stray V.* The skills that employers look for in software testers // *Software Quality Journal*. – 2019. – Vol. 27. – P. 1449–1479.

18. *Макашов П.А., Романенко Н.А.* Сервис-ориентированный подход к управлению ИТ проектами на примере использования программного продукта JIRA // *Современные информационные технологии и ИТ-образование*. – 2015. – № 2. – С. 127–132.

19. *Ясько А.О.* Обзор ITSM-систем для автоматизации бизнес-процессов внедрения и сопровождения программного обеспечения // *Научный потенциал молодежи и технический прогресс*. – 2018. – № 1. – С. 18–20.

20. *Zolkifli N.N., Ngah A., Deraman A.* Version control system: a review // *Lecture Notes in Computer Science*. – 2018. – Vol. 135. – P. 408–415.

Гаврильев Эрчимэн Иванович, аспирант кафедры теоретической и прикладной информатики Новосибирского государственного технического университета. E-mail: erchimen_gavriliev@outlook.com

Авдеенко Татьяна Владимировна, профессор кафедры теоретической и прикладной информатики Новосибирского государственного технического университета. E-mail: tavdeenko@mail.ru

DOI: 10.17212/2782-2230-2021-2-67-84

Model for assessing the qualification of it specialists in software development*

E.I. Gavriliev¹, T.V. Avdeenko²

¹ *Novosibirsk State Technical University, 20 Karl Marx Prospekt, Novosibirsk, 630073, Russian Federation, PhD student of the Department of Theoretical and Applied Computer Science. E-mail: Erchimen_gavriliev@outlook.com*

² *Novosibirsk State Technical University, 20 Karl Marx Prospekt, Novosibirsk, 630073, Russian Federation, professor of the Department of Theoretical and Applied Computer Science. E-mail: tvadveenko@mail.ru*

The decisive factor for the successful work of modern IT organizations is the professional qualities of the personnel and the system for its development. Staff training for software development organizations is of particular importance due to the fact that activities in this field

* Received 19 April 2021.

make high demands on the level of qualifications of employees, their knowledge and skills, while they tend to quickly lose their relevance. However, there are problems in the assessment of personnel qualifications that can affect its' objectivity and the effectiveness of further management decisions, for example: favoritism, bias, high exactingness, an erroneous standard. The purpose of the work is to develop a model for assessing the qualifications of IT specialists in the field of software development. . This study investigates personnel assessment system, types of criteria for assessing the qualifications of software developers and testers, which cover technical aspects of activities, interpersonal and mental characteristics of specialists. To assess the qualifications, a model of the conceptual theory of software development competence was selected and changed. Based on a literature review, analysis of vacancies and interviews with IT specialists, the main areas of activity of developers and testers were identified, as well as characteristics that need to be assessed. Based on the modified model, a prototype of the system was developed in the JavaScript programming language and on the Node JS software platform for collecting information, which loads data from the Atlassian Jira version control systems for further evaluation.

Keywords: software developers, software testers, competencies, personnel assessment methods, personnel assessment criteria, personnel management, career management, process theory, Node.JS

REFERENCES

1. Nosyreva I.G. Analiz effektivnosti sistemy otsenki personala [Analysis of the effectiveness of the personnel appraisal system]. *Ekonomika truda = Labor economics*, 2019, no. 1 (6), pp. 440–452.
2. Kibanov A. Ja. Upravlenie personalom v Rossii teorija otechestvennaja i zarubezhnaja praktika [Human Resource Management in Russia: Theory, Domestic and Foreign Practice]. Moscow, INFRA-M, 2014. 283 p.
3. Dzhincharadze G.R. Protsedura otsenki personala v ramkakh sistemy upravleniya chelovecheskimi resursami [Personnel assessment procedure within the framework of the human resource management system]. *Naukovedenie = Science of Science*, 2013, no. 3, pp. 1–12.
4. Gafurova O.Yu. Kachestvennye, kolichestvennye i kombinirovannye metody otsenki personala [Qualitative quantitative and combined methods of personnel assessment]. *Aktual'nye problemy gumanitarnykh i estestvennykh nauk = Actual problems of the humanities and natural sciences*, 2017, no. 5 (4), pp. 55–58.
5. Surakka S. What subjects and skills are important for software developers? *Communications of the ACM*, 2007, vol. 50 (1), pp. 73–80.
6. Robillard M.P., Coelho W., Murphy G.C. How effective developers investigate source code: an exploratory study *IEEE Transactions on Software Engineering*, 2004, vol. 30 (12), pp. 889–903.
7. Ahmadzadeh M. An analysis of patterns of debugging among novice computer science students. *ACM SIGCSE Bulletin*, 2005, vol. 37 (3), pp. 84–88.

8. Ahmed F., Fernando Capretz L., Bouktif S., Campbell P. Soft skills requirements in software development jobs: a cross cultural empirical study. *Journal of Systems and Information Technology*, 2005, vol. 14 (1), pp. 58–81.
9. Hewner M., Guzdiel M. What game developers look for in a new graduate: interviews and surveys at one game company. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. New York, 2010, pp. 275–279.
10. Begel A., Simon B. Novice software developers, all over again. *Computing Education Research Workshop*, Sydney, Australia, 2008, pp. 3–14.
11. Li P.L., Ko A.J., Begel A. What distinguishes great software engineers? *Empirical Software Engineering*, 2020, vol. 25, pp. 322–352.
12. Li P.L., Ko A.J., Zhu J. What makes a great software engineer? *37th IEEE International Conference on Software Engineering*, 2015, vol. 1, pp. 700–711.
13. Baltes S., Diehl S. Towards a theory of software development expertise. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018*. ACM, Lake Buena Vista, FL, 2018, pp. 187–200.
14. *Certified Software Tester (CSTE)*: website. Available at: <https://www.soft-warecertifications.org/cste/> (accessed 30.05.2021).
15. Iivonen J., Mäntylä M., Itkonen J. Characteristics of high performing testers: a case study. *ESEM' 10: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, Bolzano-Bozen, Italy, 2010, pp. 1–9.
16. Deak A. What characterizes a good software tester? – A survey in Four Norwegian Companies. *Testing Software and Systems. ICTSS 2014. Lecture Notes in Computer Science*. Berlin, Heidelberg, Springer, 2014, vol. 8763, pp. 161–172. DOI: 10.1007/978-3-662-44857-1_11.
17. Florea R., Stray V. The skills that employers look for in software testers. *Software Quality Journal*, 2019, vol. 27, pp. 1449–1479.
18. Makashov P.A., Romanenko N.A. Servis-orientirovannyi podkhod k upravleniyu IT proektami na primere ispol'zovaniya programmnogo produkta JIRA [Service-oriented approach to IT project management through the example of using the JIRA software product]. *Sovremennye informatsionnye tekhnologii i IT-obrazovanie = Modern information technology and IT education*, 2015, no. 2, pp. 127–132.
19. Yas'ko A.O. Obzor ITSM-sistem dlya avtomatizatsii biznes-protsessov vnedreniya i soprovozhdeniya programmnogo obespecheniya [Review of ITSM systems for automating business processes of software implementation and maintenance]. *Nauchnyi potentsial molodezhi i tekhnicheskii progress = Scientific potential of youth and technological progress*, 2018, no. 1, pp. 18–20.

20. Zolkifli N.N., Ngah A., Deraman A. Version control system: a review. *Lecture Notes in Computer Science*, 2018, vol. 135, pp. 408–415.

Для цитирования:

Гаврильев Э.И., Авдеенко Т.В. Модель оценки квалификации ИТ-специалистов в области разработки программного обеспечения // Безопасность цифровых технологий. – 2021. – № 2 (101). – С. 67–84. – DOI: 10.17212/2782-2230-2021-2-67-84.

For citation:

Gavriliev E.I., Avdeenko T.V. K modeli otsenki kvalifikatsii IT-spetsialistov v oblasti razrabotki programmno obespecheniya [Model for assessing the qualification of it specialists in software development]. *Bezopasnost' tsifrovyykh tekhnologii = Digital technology security*, 2021, no. 2 (101), pp. 67–84. DOI: 10.17212/2782-2230-2021-2-67-84.