

Структурная модель оценки квалификации ИТ-специалистов на основе интеллектуального анализа данных информационных систем

Э. И. Гаврильев, Т. В. Авдееenko

Новосибирский государственный технический университет (НГТУ)

Аннотация: Ключевыми факторами достижения успеха проекта ИТ-организации являются персонал и его профессиональные качества. Для выявления направлений карьерного роста руководство организаций проводит перманентное оценивание квалификации сотрудников. Однако низкий уровень надежности и точности результатов оценивания, связанный с субъективным мнением руководителей, может привести к негативным последствиям принимаемых на его основе решений. Для повышения точности результатов и выявления дополнительных критериев профессиональной компетентности были разработаны модели оценки квалификации разработчиков и тестировщиков ПО для поддержки принятия решений на основе большого количества показателей, полученных при помощи интеллектуального анализа данных информационных систем, в которых работают ИТ-специалисты. В настоящей работе был проведен факторный анализ для выявления структуры пространства показателей и обнаружения в них скрытых закономерностей. Результаты показали, что при оценивании квалификации ИТ-специалистов руководители учитывают факторы, связанные с решением технических задач, результативностью и оперативностью тестирования, а также коммуникативными навыками.

Ключевые слова: оценка персонала, разработчик, тестировщик, исследовательский факторный анализ, подтверждающий факторный анализ, интеллектуальный анализ данных.

Для цитирования: Гаврильев Э. И., Авдееenko Т. В. Структурная модель оценки квалификации ИТ-специалистов на основе интеллектуального анализа данных информационных систем // Вестник СибГУТИ. 2023. Т. 17, № 4. С. 15–33.
<https://doi.org/10.55648/1998-6920-2023-17-4-15-33>.



Контент доступен под лицензией
Creative Commons Attribution 4.0
License

© Гаврильев Э. И., Авдееenko Т. В., 2023

Статья поступила в редакцию 29.04.2023;
переработанный вариант – 14.05.2023;
принята к публикации 15.05.2023.

1. Введение

Инновации в ИТ-области считаются определяющими факторами изменений в экономической и социальной сферах. Например, информатизация и автоматизация могут существенно поменять процессы и индивидуальные профили должностей во многих отраслях, например, в здравоохранении [1].

Технологические инновации влияют также и на саму ИТ-индустрию: рост сложности разрабатываемых информационных систем, а также широкий спектр автоматизируемых областей требуют от ИТ-организаций специалистов с высоким уровнем профессиональной компетентности. В связи с этим особую актуальность приобретает система профессионального развития сотрудников. Её важным элементом является оценка персонала, на основе ко-

торой принимается множество управленческих решений: повышение, понижение, перевод кадров, расторжение трудового договора.

Оценка квалификации сотрудников чаще всего базируется на субъективном мнении руководителей. Этот способ подвергается критике из-за низкого уровня надежности и точности [2–4]. Некорректная оценка сотрудника может негативно повлиять на его дальнейшее профессиональное развитие.

В предыдущих работах авторов [5, 6] для повышения точности и выявления дополнительных критериев профессиональной компетентности тестируемых и разработчиков ПО были разработаны система поддержки принятия решений (СППР), процедура и модели оценивания квалификации. Модели оценивания включали большое количество показателей из информационных систем, в которых работают ИТ-специалисты, что увеличивает длительность и трудоемкость процесса оценивания. Поэтому целью данной работы является упрощение процедуры оценивания и повышение её точности за счет применения методов факторного анализа.

Настоящая работа состоит из следующих разделов. В разделе 2 описаны результаты анализа актуальных работ, связанных с методами и критериями оценивания квалификации ИТ-специалистов. В разделе 3 представлен подход к исследованию и описаны особенности организации, в рамках которой проводилось исследование. Раздел 4 содержит результаты исследования и их интерпретацию, и последний раздел – заключение.

2. Обзор актуальных работ

Вопрос оценивания квалификации разработчиков чаще всего рассматривался со стороны технических навыков и знаний, в основном связанных с программированием [7, 8]. Несмотря на это, часть исследований предоставляет целостное видение их компетенций, объединяющее технические и гибкие навыки.

В исследовании [9] был проведен опрос сотрудников Microsoft о характеристиках «великого» разработчика ПО. На основе результатов опроса было выделено 54 атрибута, классифицированных по 4 группам: «Личностные характеристики», «Принятие решений», «Работа с командой» и «Программный продукт». Наиболее высокий уровень важности получили атрибуты, связанные с разработкой ПО, а также индивидуальными способностями самого разработчика – например, обращение внимания на качество исходного кода, наличие достаточных умственных способностей для решения комплексных задач и хорошая осведомленность о заказчиках и продуктах компании.

В работе [10] представлена концептуальная теория компетентности в области разработки программного обеспечения (SDExp), основанная на ответах опроса 355 разработчиков ПО и литературе по тематике экспертных знаний и производительности труда. Было отмечено, что результативность разработчика зависит от уровня его знаний и опыта, а также ряда индивидуальных особенностей.

Следует отметить, что существующие публикации, изучающие методы оценивания квалификации разработчиков, в основном сфокусированы на программировании [11]. Однако они могут оказаться неэффективными при оценивании квалификации в целом, поскольку разработчики диверсифицировали свою деятельность по другим направлениям: сбор и анализ требований, тестирование, проектирование и т.д.

Подходы к оцениванию квалификации тестируемых ПО рассматривались в литературе довольно фрагментарно, в отличие от подходов к оцениванию квалификации разработчиков ПО. Часть имеющихся публикаций изучает требования к тестируемым для трудоустройства [12, 13]. Например, они должны уметь управлять процессом тестирования, писать тест-кейсы и разрабатывать автотесты. Другая часть работ рассматривает знания и навыки «эффективных» и «хороших» тестируемых [14, 15]. В них упоминаются коммуникативные навыки, технические навыки (программирование, администрирование операционных систем)

и навыки тестирования, однако отсутствуют убедительные свидетельства того, что эти характеристики можно использовать при оценивании квалификации сотрудника.

В целом в имеющихся исследованиях не проанализированы методы, которые можно применить для оценивания навыков и знаний тестировщика. Метод, основанный на субъективном мнении непосредственного руководителя, может предоставить некорректные результаты ввиду следующих факторов: наличие особенностей во взаимоотношениях между руководителем и подчиненным; высокая требовательность руководителя; эффект края, при котором учитывается только последняя неделя работы и т.д.

Однако в отдельных публикациях для оценивания результатов деятельности ИТ-специалистов применяется интеллектуальный анализ данных из репозитория проектов, над которыми они работали [16]. В них было продемонстрировано, что специалист с большим уровнем вклада в проект может иметь более высокий уровень «качества».

Из анализа имеющейся литературы можно заключить, что метод оценивания сотрудника, основанный на использовании данных из систем, в которых он работает, отличается более высоким уровнем объективности, чем мнение непосредственного руководителя, т.к. в основе этого метода лежат количественно измеряемые показатели и данные, соответствующие действительности.

3. Процедура сбора данных для оценки квалификации ИТ-специалистов

В своей ежедневной работе ИТ-специалисты используют системы контроля версий, управления задачами, управления тестированием и управления знаниями [17]. Система контроля версий (Version Control System) предназначена для фиксирования изменений в исходном коде приложения [18]. Система управления задачами (Issue Tracking System) используется для организации совместной деятельности проектной команды [19]. В этой системе проекты декомпозируются на задачи, например: исправление дефекта или проектирование интерфейса. Система управления тестированием (Test Management System) используется для администрирования и контроля процессов тестирования, разработки и хранения тест-кейсов и тест-планов [20]. Система управления знаниями (Knowledge Management System) применяется для поддержки создания, хранения и передачи знаний [21]. В ней, например, содержится проектная и техническая документация.

Информацию из этих систем можно обработать при помощи методов и инструментов интеллектуального анализа данных, чтобы оценить уровень квалификации ИТ-специалиста. На основе обзора актуальных работ были выделены показатели для расчета оценки профессиональных навыков и знаний тестировщиков и разработчиков. Например, информация из системы управления задачами используется для расчета показателей, связанных с оперативностью решения задач:

- среднее количество решенных задач в день;
- среднее количество задач, решенных в рамках и вне рамок оценки;
- количество созданных задач с типом «Ошибка» с приоритетами «Blocker», «Critical», «Major», «Minor».

Данные из системы управления тестированием используются для расчета показателей производительности разработки тест-кейсов:

- количество созданных тест-кейсов;
- среднее количество шагов в тест-кейсах;
- количество выполненных тестов в статусах «Пройден», «Заблокирован», «Провален» и «Не запускался».

Информация из системы контроля версий используется для расчета показателей сложности написанного исходного кода:

- уровень поддерживаемости кода;

- цикломатическая сложность;
- меры сложности Холстеда.

Сведения из системы управления задачами используются для расчета показателей активности сотрудника, таких как количество созданных страниц и количество обновлений содержимого страниц.

Для проведения исследований был разработан прототип системы поддержки принятия решений на Node.js [5]. Прототип СППР состоит из двух подсистем: подсистемы загрузки данных из внешних систем и подсистемы оценивания сотрудника. Первая подсистема проводит миграцию информации из указанных выше систем в конце календарного дня, а вторая подсистема выполняет саму процедуру оценивания.

Разработанные СППР и реализованные в них модели принятия решений были внедрены в компанию среднего размера, которая разрабатывает ПО для банковской отрасли. Для разработки программных продуктов компании используется собственный low-code конструктор приложений. Оценивание профессиональных знаний и навыков сотрудников проводится непосредственными руководителями на основе 8 балльной шкалы, где 1 – это минимальное значение, а 8 – максимальное. Atlassian Jira используется как система управления задачами, Atlassian Confluence – как система управления знаниями, TestLink – как система управления тестированием, а в качестве системы контроля версий выступает собственный конструктор приложений.

После выполнения первичных расчетов и анализа результатов был проведен ряд интервью с ведущими специалистами и руководителями компании для корректировки набора показателей оценки. В результате для оценки квалификации тестирующих был добавлен ряд показателей [22]:

- показатели оценки качества отчета о дефекте: индексы удобочитаемости описания дефекта, наличие фактического и ожидаемого результатов;
- показатели, связанные с результатом решения зарегистрированного дефекта: процент задач, решенных с резолюциями «Решено», «Не могу воспроизвести», «Дубликат»;
- показатели оценки производительности тестирования: среднее время тестирования задачи в минутах, среднее количество проверенных задач в день;
- показатели, относящиеся к менторству новых сотрудников: количество подключений к менторству, количество успешно закрытых стажировок.

Для оценки квалификации разработчиков были добавлены:

- показатели для подсчета количества объектов, разработанных в конструкторе приложений: количество форм, функций, процессов, SQL-запросов и т.д.;
- показатели для подсчета применяемых элементов языка программирования: количество применений Java-библиотек, функций для интеграции через JDBC- и HTTP-протоколы, функций для генерации запросов в форматах XML и JSON;
- показатели для подсчета упоминаний применяемых инструментов и технологий, а также типов решаемых задач.

Для подсчета количества упоминаний был использован инструмент Томи-парсер, который проводит обработку текстов на естественном языке и извлекает из них факты с учетом синтаксиса языка и морфологии [23]. Факты извлекаются из описаний задач, комментариев и журналов работ по задачам, а для их извлечения были подготовлены наборы ключевых слов, в которых перечислены различные варианты формулировок факта.

Например, для извлечения упоминаний XSD и WSDL в описаниях задач были подготовлены 2 правила, на основе которых парсер ищет согласованные по падежу слова, одно из которых должно упоминаться в наборе ключевых слов (Рис. 1).

```
// поиск XSD
S -> Word<c-agr[1]>* Word<kwset=[xsd], c-agr[1], rt> interp (DevTechXSD.Term::not_norm);

// поиск WSDL
S -> Word<c-agr[1]>* Word<kwset=[wsdl], c-agr[1], rt> interp (DevTechWSDL.Term::not_norm);
```

Рис. 1. Примеры правил для извлечения фактов

Составленные наборы ключевых слов могут включать в себя названия технологий и инструментов на английском и русском языках (Рис. 2).

```
TAuxDicArticle wsdl
{
    key = "wsdl"
    key = "всдл"
}

TAuxDicArticle xsd
{
    key = "xsd"
    key = "хсд"
}
```

Рис. 2. Пример словаря ключевых слов

В итоге для оценки профессиональных знаний и навыков тестировщика было выделено 50 показателей, а для оценки разработчика – 115 показателей. Начальная версия выборки включала в себя 66 результатов оценивания квалификации 25 тестировщиков и 494 результата оценивания 142 разработчиков в период с 2019–2022 гг. Для каждого результата был определен интервал дат, за который необходимо провести выгрузку данных из систем и расчет показателей. В итоге была загружена информация по 218167 задачам, 4206 страницам из системы управления знаниями, 75576 тест-кейсам из системы управления тестированием и был выгружен код из 39 репозиторийев.

При оценивании квалификации использовалось большое количество показателей, которые можно объединить в факторы для упрощения процедуры и сокращения размерности данных, а также рассмотреть неявные закономерности и связи между показателями для более точных результатов. Для достижения этой цели были применены методы исследовательского факторного анализа (Exploratory Factor Analysis, EFA) и структурного моделирования (Structural Equation Modeling, SEM).

Перед проведением факторного анализа был проведен поиск выбросов на основе графика boxplot, из исходных данных результатов оценивания квалификации тестировщика были удалены показатели и сами выбросы (Рис. 3).

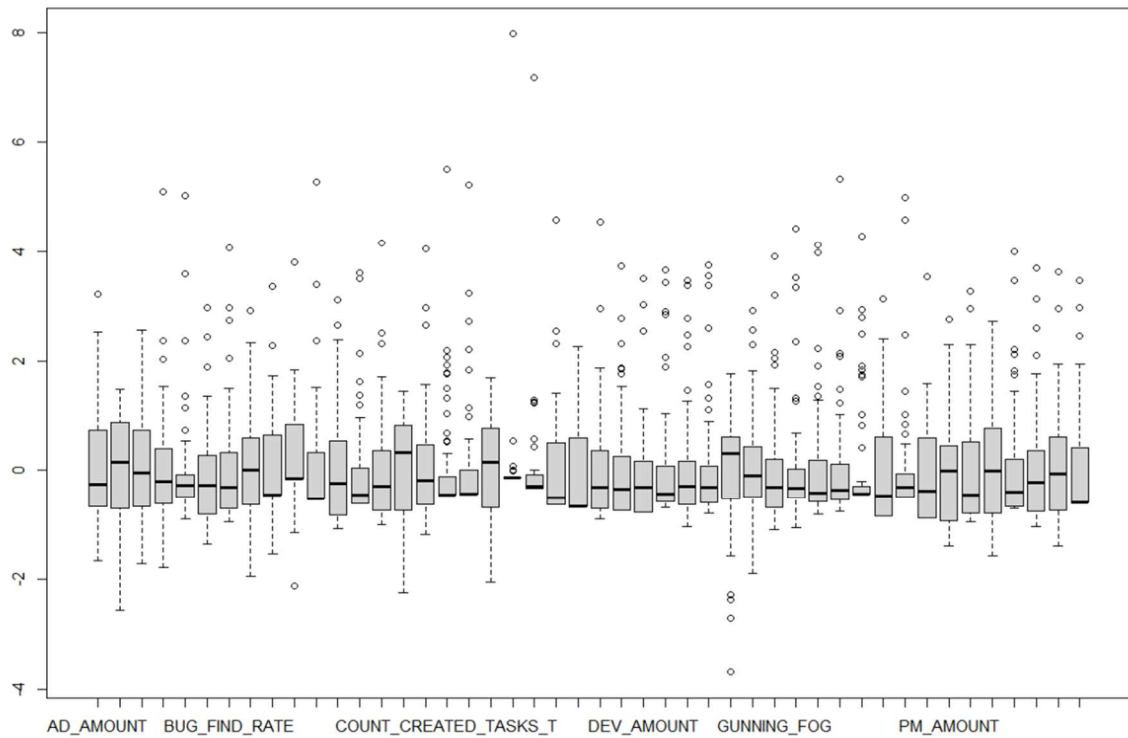


Рис. 3. График boxplot для стандартизованных значений показателей оценки квалификации тестировщика

При поиске аномалий в исходных данных оценок разработчиков было выявлено, что лишними являются результаты оценивания квалификации для руководителей команд разработки, которые в большей степени занимаются управленческими задачами (Рис. 4). После их удаления конечная версия выборки включала 237 результатов по 70 разработчикам.

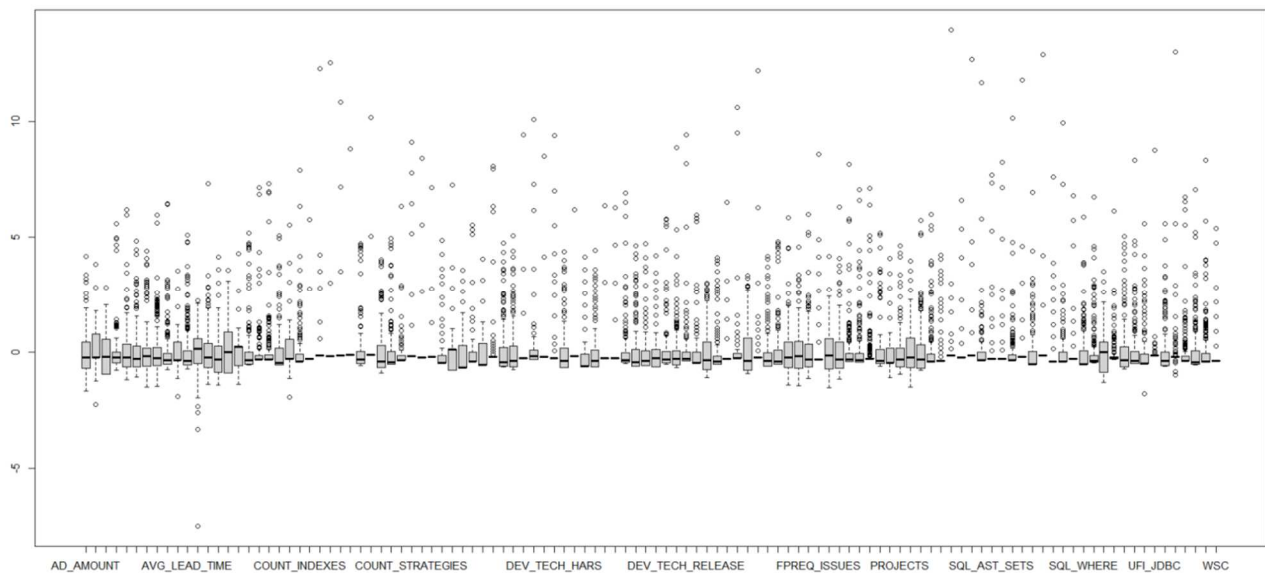


Рис. 4. График boxplot для стандартизованных значений показателей оценки квалификации разработчика

4. Факторный анализ

4.1. Анализ модели оценки профессиональных навыков и знаний тестирующих

Для проведения исследовательского факторного анализа был использован метод главных компонент (Principal Component Analysis, PCA). Вначале была проведена оценка целесообразности проведения факторного анализа при помощи критериев Кайзера–Мейера–Олкина (КМО) и сферичности Бартлетта. Результаты теста Бартлетта указывали на хорошие перспективы анализа – 181.9 и $p\text{-value} < 2.2e-16$, но начальное значение теста КМО было низким, поэтому из модели были удалены показатели, у которых значение этого критерия было меньше критического (< 0.6) [24]. После их удаления значение критерия КМО было равно 0.74, что указывает на хорошие перспективы проведения факторного анализа.

Из графика «каменистой осыпи» на рис. 5 было определено возможное число факторов – оно варьировалось между 2 и 3. Факторный анализ был проведен при помощи функции базового пакета R – `factanal`. Итоговая версия модели состояла из 5 факторов, т.к. при её построении был получен наибольший $p\text{-value}$ гипотезы достаточности факторов по сравнению с моделями из 2, 3 и 4 факторов, равный 0.263.

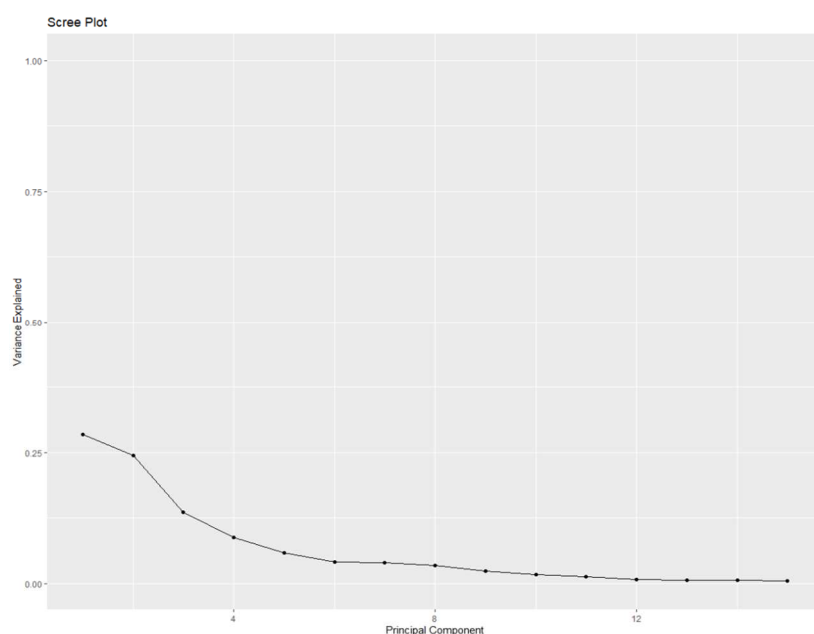


Рис. 5. График «Каменистая осыпь» для модели оценки квалификации тестирующих

Нагрузки факторов, полученных при помощи метода вращения `varimax`, представлены в табл. 1, абсолютные значения меньше 0.4 не были включены в факторную модель.

Таблица 1. Таблица нагрузок для факторов оценки квалификации тестирующего

	F1	F2	F3	F4	F5
AUTOMATED_READABILITY		0.934			
BUG_FIND_RATE	0.668				
COLEMAN_LIAU		0.923			
COUNT_AVG_TEST_BY_DATE	0.891				
COUNT_BUGS_BY_DATE					0.649
COUNT_CREATED_TASKS_B			0.430		
COUNT_PLATF_UPD				0.627	
COUNT_REOPENS_BY_USER	0.954				
FIXED_ISSUES			0.789		

FIXED_ISSUES_NOT_WITHIN_EST			0.752		
FIXED_ISSUES_WITHIN_EST			0.968		
FLESCH_KINCAID	0.167	0.921			
LOG_FILES				0.708	
MULTIPLE_CASES	0.587				
PASSED_TRAINEE_AMOUNT				0.820	

Доля общей дисперсии, объясняемой каждым фактором, и накопление этих дисперсий представлены в табл. 2.

Таблица 2. Доля общей дисперсии и её накопление для факторов оценки квалификации тестирующего

	F1	F2	F3	F4	F5
SS loadings	2.972	2.851	2.491	1.814	0.690
Proportion Var	0.198	0.190	0.166	0.121	0.046
Cumulative Var	0.198	0.388	0.554	0.675	0.721

Полученную модель факторов можно интерпретировать следующим образом:

Фактор F1 – «Результативность тестирования». Включает в себя показатели:

- скорость обнаружения ошибок (BUG_FIND_RATE);
- количество зарегистрированных дефектов с несколькими кейсами воспроизведения ошибок (MULTIPLE_CASES);
- количество проверок задач в день (COUNT_AVG_TEST_BY_DATE);
- количество повторных открытий задач, инициированных сотрудником (COUNT_REOPENS_BY_USER).

Данный фактор характеризует результативность основной деятельности ИТ-специалиста – тестирование. От объема проверяемых сотрудником задач зависит уровень стабильности информационной системы. Каждый обнаруженный дефект или задание по разработке функционального элемента программного продукта регистрируется в системе управления задачами. После исправления дефекта или завершения разработки сотрудник изменяет статус задачи на «Решено», чтобы тестирующий начал проверку. После окончания проверки тестирующий переоткрывает задачу, если была обнаружена ошибка, или меняет её статус на «Проверено», если ошибки не были найдены.

Фактор F2 – «Письменные коммуникативные навыки». Содержит показатели:

- среднее значение индекса автоматической читаемости описания дефекта (AUTOMATED_READABILITY);
- среднее значение индекса Колмана–Лиау описания дефекта (COLEMAN_LIAU);
- среднее значение индекса Флеша–Кинкейда описания дефекта (FLESCH_KINCAID).

Этот фактор иллюстрирует степень легкости восприятия составленных описаний дефектов. Чем доступнее и понятнее отчет о дефекте, тем быстрее и эффективнее его сможет решить разработчик.

Фактор F3 – «Решение задач в рамках оценки». Объединяет переменные:

- количество решенных задач тестирующим (FIXED_ISSUES);
- количество решенных задач вне рамок оценки (FIXED_ISSUES_NOT_WITHIN_EST);
- количество решенных задач в рамках оценки (FIXED_ISSUES_WITHIN_EST);
- процент зарегистрированных дефектов с приоритетом «Blocker» (COUNT_CREATED_TASK_B).

Текущий фактор описывает, насколько успешно сотрудник решает задачи в соответствии с оцененным количеством времени. Если задачи не выполняются за оцененное количество времени, то повышаются расходы проекта и риск срыва сроков его сдачи.

Фактор F4 – «Технические инструменты и наставничество». Состоит из показателей:

- количество упоминаний обновления версий конструктора приложений в задачах, журнале работ (COUNT_PLATF_UPD);
- количество вложений, связанных с журналированием работы системы (LOG_FILES);
- количество стажеров, успешно прошедших стажировку под менторством сотрудника (PASSED_TRAINEE_AMOUNT).

Фактор характеризует степень владения техническими инструментами, используемыми при тестировании, а также навыки наставничества. В процессе проверки системы тестирующим требуется дополнительно собирать журналы работы сервера и архив запросов, чтобы разработчик оперативнее смог определить причину дефекта.

Иногда причиной дефекта является ошибка в работе конструктора приложений. Поэтому после исправления ошибки в новой версии конструктора приложений тестирующие устанавливают её, чтобы проверить актуальность дефекта. Сам процесс обновления версии конструктора требует от сотрудника знаний и навыков по администрированию сервера приложений и операционной системы, т.к. шаги по установке являются нетривиальными.

Наличие показателя, связанного с количеством стажеров, связано с тем, что в исследуемой компании более компетентным тестирующим предлагают обучить новых сотрудников.

Фактор F5 – «Оперативность обнаружения дефектов». Последний фактор определяет, насколько оперативно тестирующий обнаруживает дефекты, он состоит из 1 показателя:

- количество зарегистрированных дефектов в день (COUNT_BUGS_BY_DATE).

Для проверки того, насколько точно модель соответствует данным, был проведен подтверждающий факторный анализ (Confirmatory Factor Analysis, CFA). В рамках него были рассчитаны 3 показателя, их значения указывают на относительно приемлемое качество модели:

- CFI (сравнительный индекс соответствия, > 0.950) – 0.894;
- RMSEA (среднеквадратичная ошибка аппроксимации, < 0.1) – 0.116;
- SRMR (стандартизованный среднеквадратичный остаток, < 0.1) – 0.105.

На основе результатов исследовательского факторного анализа была построена структурная модель зависимости между оценкой профессиональных навыков и знаний, полученной от руководителя, и выделенными факторами (Рис. 6). Для разработки структурной модели был использован пакет lavaan в R.

В рамках валидации структурной модели состав её факторов был изменен:

1. Из модели были удалены избыточные факторы, решение об их исключении принималось на основе p-value из коэффициентов структурной модели, его значение должно быть меньше 0.05.
2. Дополнительно были добавлены 2 фактора:

Фактор F6 «Инструменты тестирования веб-сервисов». Данный фактор характеризует степень владения сторонними инструментами для тестирования работы SOAP и REST веб-сервисов. Он включает в себя показатель – количество упоминаний инструментов тестирования веб-сервисов в описаниях задач, комментариях и журналах работ (COUNT_INTEGRATION_INSTR).

Фактор F7 «Длительность решения задач». Данный фактор характеризует объем трудозатрат тестирующего на решение назначенных задач. Состоит из показателя – общее время на решение задач (ALL_ISSUE_RES_TIME).

Эти показатели имели высокую степень значимости в регрессионной модели оценки профессиональных знаний и навыков, но они были исключены в рамках EFA, т.к. имели низкое значение индекса КМО.

Параметры оценки структурной модели также свидетельствуют об относительно приемлемом качестве:

- количество параметров – 17;
- степени свободы – 11;
- χ^2 – 20.197;
- CFI – 0.948;

– RMSEA – 0.113;

– SRMR – 0.054.

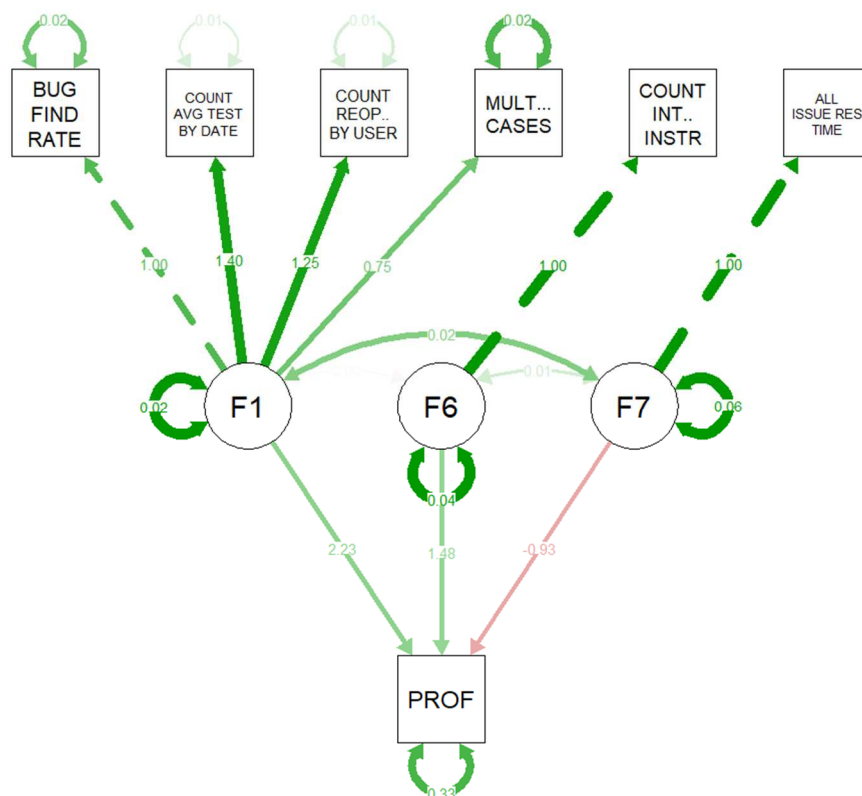


Рис. 6. Структурная модель оценки профессиональных знаний и навыков тестировщика

Полученные коэффициенты для разработанной структурной модели представлены в табл. 3.

Таблица 3. Коэффициенты структурной модели оценки профессиональных знаний и навыков тестировщика

PROF ~	Estimate	Std.Err	t-value	P(> t)
F1	2.230	0.718	3.106	0.002
F6	1.477	0.386	3.829	0.000
F7	-0.927	0.345	-2.688	0.007

По результатам построенной модели можно увидеть, что фактор F1 «Результативность тестирования» имеет положительное влияние на оценку, т.к. тестирование является основной должностной обязанностью работников. Чем больше изменений в новой версии программного продукта будет проверено, тем выше становится уровень её стабильности и надежности.

Знания и опыт работы с техническими инструментами позволяют провести более комплексное и тщательное тестирование программных продуктов, поэтому положительно на оценку влияет фактор F6 «Инструменты тестирования веб-сервисов». Сотрудники в исследуемой компании используют такие специальные пакеты ПО, как Soap UI, Postman и JMeter, т.к. они предоставляют тестировщикам возможность разработать автотесты, провести нагрузочное тестирование, а также собрать подробную статистику о пройденных и не пройденных тест-кейсах.

Негативно на оценку уровня профессиональных навыков влияет фактор F7 «Длительность решения задачи». Данный фактор тесно связан с бюджетом и сроком проекта: чем дольше решаются задачи, тем выше становится уровень операционных расходов и риск срыва установленных сроков. Более компетентные тестировщики за меньшее количество времени решают назначенные на них задачи.

4.2. Анализ модели оценки профессиональных навыков и знаний разработчиков

Для модели оценки профессиональных навыков и знаний разработчиков результаты теста Бартлетта указывали на хорошую возможность выполнения анализа – 436, $p\text{-value} < 2e-16$, но начальное значение теста КМО было средним, в связи с чем из модели были удалены избыточные показатели. После их удаления значение критерия КМО было равно 0.87.

Построенный график «Каменистая осыпь» свидетельствует о том, что подходящее число факторов варьировалось между 2 и 3 (Рис. 7).

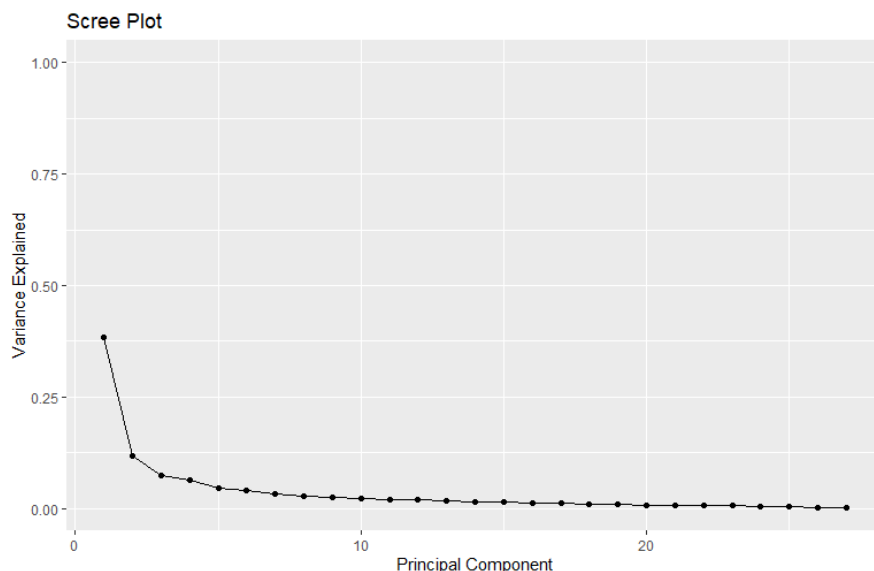


Рис. 7. График «Каменистая осыпь» для модели оценки квалификации разработчиков

После выполнения EFA и построения структур факторов был проведен CFA. Показатели указывали на удовлетворительную состоятельность моделей. Наибольшие значения показателей были получены для модели, состоящей из 4 факторов, по сравнению с моделями, состоящими из 2 и 3 факторов: CFI – 0.691, RMSEA – 0.143, SRMR – 0.112.

В результате корректировок набора показателей была построена новая версия модели, состоящая из 5 факторов, т.к. показатели CFA для этого количества факторов были лучше, чем для меньшего количества факторов. Рассчитанные при помощи метода varimax нагрузки факторов представлены в табл. 4.

Таблица 4. Таблица нагрузок для факторов оценки квалификации разработчика

	RC1	RC2	RC3	RC4	RC5
AVG_ISSUES_BY_WL				0.825	
COUNT_ENV_PARAMS		0.590			
COUNT_JS_CLIENT_LIBS			0.820		
COUNT_MC_CONSTRUCTOR		0.753			
COUNT_MC_SQL		0.858			
COUNT_XML_COMPS	0.820				
DEV_TECH_CONSULTATION				0.648	
DEV_TECH_HARS			0.831		
DEV_TECH_JAVA	0.614				
DEV_TECH_XSD					0.613
FPREQ_ISSUES			0.662		
MTYPE_ATTR	0.792				
TEMP_TYPES	0.826				
UDMS					0.778

UFI_HTTP	0.694				
UFI_JL		0.736			
UFI_XML					0.613

Доля общей дисперсии, объясняемой каждым фактором, и накопление этих дисперсий представлены в табл. 5.

Таблица 5. Доля общей дисперсии и ее накопление для факторов оценки квалификации разработчика

	RC1	RC2	RC3	RC4	RC5
SS loadings	3.804	2.875	2.482	1.818	2.042
Proportion Var	0.224	0.169	0.146	0.107	0.120
Cumulative Var	0.224	0.393	0.539	0.766	0.659

Построенную модель факторов можно интерпретировать следующим образом:

Фактор RC1 – «Разработка интеграционных сервисов». Включает в себя показатели:

- количество загруженных XSD-схем (COUNT_XML_COMPS);
- количество добавленных атрибутов в таблицы (MTYPE_ATTR);
- количество созданных временных таблиц (TEMP_TYPES);
- количество применений функций для взаимодействия с внешними системами через HTTP-протокол (UFI_HTTP);
- степень частоты упоминаний Java в назначенных задачах, комментариях и журнале работ (DEV_TECH_JAVA).

Этот фактор характеризует уровень технических навыков сотрудника и опыт разработки интеграционных сервисов. Интеграционные сервисы в программных продуктах исследуемой компании работают на основе REST-ресурсов, в которых системы обмениваются между собой XML-документами. Для формирования этих документов необходимо загрузить XSD-схему, а также сгенерировать на её основе временные таблицы и атрибуты. Сформированные XML-документы отправляются в другие системы через HTTP-протокол при помощи встроенных функций.

Фактор RC2 – «Написание SQL-запросов». Содержит показатели:

- количество разработанных поисковых методов типа «Конструктор» (COUNT_MC_CONSTRUCTOR);
- количество разработанных поисковых методов типа «SQL» (COUNT_MC_SQL);
- количество применений Java-библиотек (UFI_JL);
- количество созданных параметров приложения (COUNT_ENV_PARAMS).

Данный фактор также характеризует уровень технических навыков и опыт написания SQL-запросов. Для их исполнения в конструкторе приложений используются специальные объекты – поисковые методы, в которых разработчик может указать текст запроса. Еще одним распространенным способом является применение Java-библиотек для исполнения SQL-запросов через API JDBC.

Фактор RC3 – «Анализ дефектов конструктора приложений». Объединяет показатели:

- степень частоты упоминаний HAR-файлов в комментариях и журналах работ (DEV_TECH_HARS);
- количество зарегистрированных дефектов в проекте по поддержке конструктора приложений (FPREQ_ISSUES);
- количество разработанных JS-библиотек для экранных форм (COUNT_JS_CLIENT_LIBS).

Фактор характеризует степень владения инструментами для сбора отладочной информации и понимания основ работы конструктора приложений. В работе конструктора приложе-

ний могут возникать ошибки, и для их решения разработчики регистрируют дефекты в системе управления задачами, в проекте по поддержке конструктора. Для анализа дефектов требуется дополнительная информация, например, журналы работы сервера и HAR-файлы – архивы, содержащие историю отправленных запросов.

Фактор RC4 – «Коммуникативные навыки». Включает в себя 2 показателя: среднее количество задач, по которым работает сотрудник, в 1 день (AVG_ISSUES_BY_WL) и степень частоты упоминаний консультаций в журнале работ (DEV_TECH_CONSULTATION). Этот фактор характеризует устные коммуникативные навыки разработчика. Разработчики консультируют заказчиков и сотрудников компаний. Сами консультации запрашиваются в рамках созданных задач, поэтому разработчики с большим количеством проработанных задач в день, как правило, отвечают на вопросы.

Фактор RC5 – «Мэппинг данных в интеграционных сервисах». Состоит из показателей:

- степень частоты упоминаний XSD в задачах, комментариях и журнале работ (DEV_TECH_XSD);

- количество разработанных пользовательских функций (UDMS);

- количество применений функций для генерации запросов в формате XML (UFI_XML).

Этот фактор характеризует опыт разработки обработчиков запросов и ответов из интеграционных сервисов. Мэппинг данных – процесс сопоставления атрибутов запросов и ответов из внешней системы со связанными атрибутами таблиц разрабатываемого приложения [25]. При мэппинге запросов и ответов из внешних систем разработчики используют встроенные функции по разбору XML-документа и пишут пользовательские функции для их обработки.

Результаты проведения CFA свидетельствуют об относительно приемлемом качестве модели: CFI – 0.841, RMSEA – 0.132, SRMR – 0.072.

На основе результатов EFA была разработана структурная модель зависимости между оценкой профессиональных навыков и знаний, полученной от руководителя, и выделенными факторами. В рамках валидации структурной модели из неё был удален избыточный фактор RC3, т.к. у него было высокое значение p -value – 0.845. Итоговая версия модели представлена на рис. 8.

Параметры оценки структурной модели также свидетельствуют об относительно приемлемом качестве:

- количество параметров – 39;

- степени свободы – 81;

- χ^2 – 425.515;

- CFI – 0.862;

- RMSEA – 0.134;

- SRMR – 0.064.

Полученные коэффициенты для разработанной модели представлены в табл. 6.

Таблица 6. Коэффициенты структурной модели оценки профессиональных знаний и навыков разработчика

PROF ~	Estimate	Std. Err	t-value	P(> t)
RC1	2.245	0.568	3.953	0.000
RC2	3.757	0.703	5.346	0.000
RC4	3.314	0.924	3.587	0.000
RC5	-3.747	0.657	-5.704	0.000

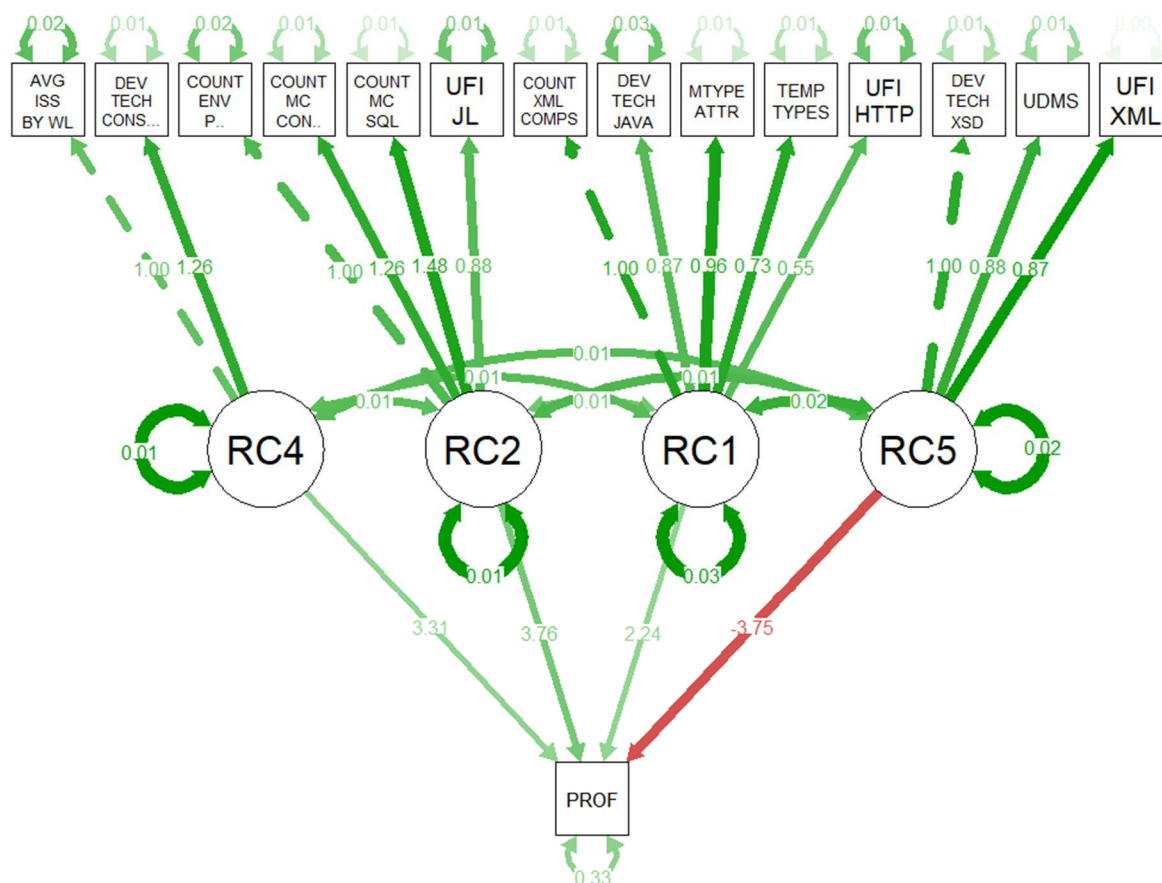


Рис. 8. Структурная модель оценки профессиональных знаний и навыков разработчика

Важными чертами квалификации разработчика являются глубокие знания инструментов для разработки и способов их применения в решении задач, поэтому положительное влияние на оценку имеют фактор RC1 «Разработка интеграционных сервисов» и фактор RC2 «Написание SQL-запросов». В большинстве случаев эти задачи требуют специфических знаний и опыта, поэтому более квалифицированные работники решают эти задачи в срок.

Коллеги, не знакомые с разрабатываемой системой, нуждаются в консультации более опытных разработчиков для выполнения назначенных задач. Также у сотрудников со стороны заказчика могут возникнуть вопросы по работе с системой и по её поддержке. В связи с чем положительное влияние на оценку оказывает фактор RC4 «Коммуникативные навыки».

Основной особенностью обмена знаниями в исследуемой компании является то, что обмен происходит в личных переписках или звонках, и знания редко фиксируются в Atlassian Confluence, поэтому показатели из этой системы оказались избыточными.

Негативное влияние на оценку имеет фактор RC5 «Маппинг данных в интеграционных сервисах». Возможно, это связано с исходными данными и невысоким уровнем сложности этой области задач: большие значения показателей данного фактора были у разработчиков, работавших в компании меньше 6 месяцев и имевших низкий уровень оценки профессиональных знаний и навыков. Чаще всего задачи, связанные с маппингом данных, назначаются на менее опытных сотрудников, т.к. они не требуют высокого уровня квалификации. В рамках этих задач разработчики пишут простой код, в котором происходит перезапись значений из одного поля в другое.

Регрессионная модель оценки из предыдущей работы [5] в большей степени рассматривала технические навыки разработчика в области клиентской части приложений (frontend), например, влиятельными оказались показатели, связанные с настройкой виджетов и количеством измененных строк кода интерфейсных форм. В то время как структурная модель учитывает технические навыки разработчика в области серверной части приложений (backend), например, факторы RC5 «Маппинг данных в интеграционных сервисах» и RC2 «Написание

SQL-запросов», но также она учитывает и нетехнические навыки, например, фактор RC4 «Коммуникативные навыки». Возможно, это связано с тем, что выборка в текущей работе включает в себя больший объем данных, а программные продукты компании за последние 2 года выступали как backend-решения, в то время как frontend-части были разработаны другими вендорами. Также разработчики начали принимать активное участие в процессах сопровождения и поддержки систем, поэтому коммуникативные навыки играют важную роль в решении задач.

При этом обе модели учитывают навыки разработчика, связанные с разработкой интеграционных сервисов: в регрессионной модели из предыдущей статьи важным оказался показатель подсчета количества применений функций для взаимодействия с внешними системами, а в структурной модели – фактор RC1 «Разработка интеграционных сервисов». Возможно, это связано с особенностями клиентов исследуемой компании: у финансовых организаций, как правило, есть комплексная ИТ-инфраструктура, в которой каждый программный продукт ответственен за исполнение ряда функций. Поэтому новые внедряемые системы должны иметь функционал для взаимодействия с другими системами заказчика для решения поставленных целей и задач.

5. Заключение

В рамках этой работы был проведен факторный анализ и построены структурные модели для оценки профессиональных знаний и навыков тестировщиков и разработчиков. Было выявлено, что при оценивании квалификации тестировщиков учитываются результативность тестирования, навыки работы с инструментами тестирования веб-сервисов и длительность решения задач. Другие факторы оценки квалификации тестировщиков, выявленные в рамках исследовательского факторного анализа, оказались избыточными по результатам разработки структурной модели. Возможно, это связано с тем, что непосредственные руководители не учитывают эти данные при оценивании квалификации. Однако факторы, связанные с письменными коммуникативными навыками, техническими инструментами для тестирования и наставничеством, рекомендуется учитывать при оценивании. Умение устно и письменно общаться с коллегами и клиентами кратким и понятным языком играет важную роль в принятии решений. Кроме того, тестировщики, выступающие в роли ментора, систематизируют и валидируют собственные знания и опыт.

При оценивании квалификации разработчиков учитываются опыт решения технических задач: в частности, написание SQL-запросов и разработка интеграционных сервисов, а также устные коммуникативные навыки.

Профили деятельности и должностные обязанности тестировщиков и разработчиков в исследуемой компании сильно отличаются, в связи с чем выявленные факторы практически не включают в себя схожие показатели.

Интеллектуальный анализ данных систем, в которых работают ИТ-специалисты, может содействовать руководству в более точном оценивании уровня квалификации сотрудника, а также помочь лучше понять критерии оценивания, процессы разработки, тестирования и сопровождения программных продуктов за счет выявления скрытых факторов и закономерностей.

При помощи разработанных моделей менеджеры могут проводить оценку квалификации тестировщика и разработчика. В дальнейшем планируется подготовить методику для составления индивидуального плана развития ИТ-специалиста в области разработки ПО. Также модели будут внедрены в подсистему оценивания СППР, чтобы на основе полученных результатов и подготовленной методики сотруднику предлагались наиболее подходящие направления профессионального роста и задачи для развития профессиональных навыков и знаний.

Литература

1. *Gastaldi L., Corso M.* Smart Healthcare Digitalization: Using ICT to Effectively Balance Exploration and Exploitation within Hospitals // *International Journal of Engineering Business Management*. 2012. V. 4. P. 1–13.
2. *Pearlman K., Schmidt F. L., Hunter J. E.* Validity generalization for tests used to predict job proficiency and training success in clerical occupations // *Journal of Applied Psychology*. 1980. V. 65, № 4, P. 373–406.
3. *Rothstein H. R.* Interrater reliability of job performance ratings: Growth to asymptote level with increasing opportunity to observe // *Journal of Applied Psychology*. 1990. V. 75, № 3. P. 322–327.
4. *Viswesvaran C., Ones D. S., Schmidt F. L.* Comparative analysis of reliability of job performance ratings // *Journal of Applied Psychology*. 1996. V. 81, № 5. P. 557–574.
5. *Gavriliev E. I., Avdeenko T. V.* Model and Procedure for Assessing the Qualification of a Software Developer // *Proc. IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM)*, 2022. P. 303–307.
6. *Гаврильев Э. И., Авдеенко Т. В.* Процедура оценки квалификации разработчика программного обеспечения // *Сборник научных трудов «Наука. Технологии. Инновации»*: в 10-и частях, Новосибирск, 06–10 декабря 2021 года. С. 145–148.
7. *Robillard M. P., Coelho W., Murphy G. C.* How effective developers investigate source code: an exploratory study // *IEEE Transactions on Software Engineering*. 2004. V. 30, № 12, P. 889–903.
8. *Surakka S.* What subjects and skills are important for software developers // *Communications of the ACM*. 2007. V. 50, №. 1. P. 73–78.
9. *Li P. L., Ko A. J., Begel A.* What Makes a Great Software Engineer? // *Proc. 37th International Conference on Software Engineering*, 2015. P. 700–710.
10. *Baltes S., Diehl S.* Towards a theory of software development expertise // *Proc. 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2018.
11. *Bergersen G. R., Sjøberg D. I. K., Dybå T.* Construction and validation of an instrument for measuring programming skill // *IEEE Trans. Softw. Eng.* 2014. V. 40, №. 12. P. 1163–1184.
12. *Juristo N., Moreno A. M., Stigel W.* Guest editors' introduction: Software testing practices in industry // *IEEE Software*. 2006. V. 23, № 4. P. 19–21.
13. *Florea R., Stray V.* The skills that employers look for in software testers // *Software Quality Journal*. 2019. № 27. P. 1449–1479.
14. *Iivonen J., Mäntylä M. V., Itkonen J.* Characteristics of high performing testers: a case // *Proc. ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2010. P. 1–9.
15. *Deak A.* What Characterizes a Good Software Tester? – A Survey in Four Norwegian Companies // *Lecture Notes in Computer Science*. 2014. № 8763. P. 162–172.
16. *Gousios G., Kalliamvakou D., Spinellis E. D.* Measuring developer contribution from software repository data // *Proc. International working conference on Mining software repositories (MSR)*, 2008. P. 129–132.
17. *Lanubile F., Ebert C., Prikladnicki R., Herzig R. K.* A Collaboration Tools for Global Software Engineering // *IEEE Software*. 2010. V. 27, № 2. P. 52–55.
18. *Zolkifli N. N., Ngah A., Deraman A.* Version Control System: A Review // *Lecture Notes in Computer Science*. 2018. V. 135. P. 408–415.
19. *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. V. 11, № 2. С. 127–132.

20. Collins E. F., Lucena Jr. V. Software Test Automation practices in agile development environment: An industry experience report // Proc. 7th International Workshop on Automation of Software Test (AST), 2012. P. 57–65.
21. Apraci I., Al-Emran M., Al-Sharaf M. A. The impact of knowledge management practices on the acceptance of Massive Open Online Courses (MOOCs) by engineering students: A cross-cultural comparison // Telematics and Informatics. 2020. № 54. P. 1–13.
22. Гаврильев Э. И., Авдеенко Т. В. Многофакторная регрессионная модель оценки квалификации тестировщика программного обеспечения // Доклады ТУСУР. 2022. V. 25, № 4. С. 115–121.
23. Томита-парсер [Электронный ресурс]. URL: <https://tech.yandex.ru/tomita/> (дата обращения: 02.04.2023).
24. Kaiser-Meyer-Olkin (KMO) Test for Sampling Adequacy [Электронный ресурс]. URL: <https://www.statisticshowto.com/kaiser-meyer-olkin/> (дата обращения: 16.04.2023).
25. Understanding Data Mapping Tools, Process, and Techniques [Электронный ресурс]. URL: <https://www.astera.com/type/blog/understanding-data-mapping-and-its-techniques/> (дата обращения: 02.04.2023).

Гаврильев Эрчимэн Иванович

аспирант кафедры теоретической и прикладной информатики, Новосибирский государственный технический университет (НГТУ, 630073, Новосибирск, просп. К. Маркса, д. 20), e-mail: erchimen_gavriliev@outlook.com, ORCID ID: 0000-0001-7289-3969.

Авдеенко Татьяна Владимировна

д.т.н., профессор кафедры теоретической и прикладной информатики, Новосибирский государственный технический университет (НГТУ, 630073, Новосибирск, просп. К. Маркса, д. 20), e-mail: tavdeenko@mail.ru, ORCID ID: 0000-0002-8614-5934.

Авторы прочитали и одобрили окончательный вариант рукописи.

Авторы заявляют об отсутствии конфликта интересов.

Вклад соавторов: Каждый автор внес равную долю участия как во все этапы проводимого теоретического исследования, так и при написании разделов данной статьи.

**Structural Model of Indicators for IT Specialists' Qualification Assessment
Based on Data Mining of Information Systems**

Erchimen I. Gavriliev, Tatiana V. Avdeenko

Novosibirsk State Technical University (NSTU)

Abstract: The key factors for the success of an IT organization project are its people and professional quality. To meet the need for professional growth, management adheres to the constant assessment of the employee's qualification. However, the low level of reliability and accuracy of the assessment results associated with the subjective opinion of the management may lead to the negative consequences of decisions. For fast results and high accuracy of proficiency testing, assessment models and software testers have been developed to support decision-making based on a large number of results obtained using data mining of information systems in the work of IT specialists. In the present work, a factor analysis was carried out to identify detection

of such cases. The results of measurements, which are determined when evaluating the qualifications of IT specialists, cause factors associated with the occurrence of technical problems, the effectiveness and efficiency of testing as well as communication skills.

Keywords: personnel assessment, developer, tester, exploratory factor analysis, confirmatory factor analysis, data mining.

For citation: Gavriliev E. I., Avdeenko T. V. Structural model of indicators for IT specialists' qualification assessment based on data mining of information systems (in Russian). *Vestnik SibGUTI*, 2023, vol. 17, no. 4, pp. 15-33. <https://doi.org/10.55648/1998-6920-2023-17-4-15-33>.



Content is available under the license
Creative Commons Attribution 4.0
License

© Gavriliev E. I., Avdeenko T. V., 2023

The article was submitted: 29.04.2023;
revised version: 14.05.2023;
accepted for publication 15.05.2023.

References

1. Gastaldi L., Corso M. Smart Healthcare Digitalization: Using ICT to Effectively Balance Exploration and Exploitation within Hospitals. *International Journal of Engineering Business Management*, 2012, vol. 4, pp. 1-13.
2. Pearlman, K., Schmidt, F. L., Hunter, J. E. Validity generalization for tests used to predict job proficiency and training success in clerical occupations. *Journal of Applied Psychology*, 1980, vol. 65, no 4, pp. 373-406.
3. Rothstein H. R. Interrater reliability of job performance ratings: Growth to asymptote level with increasing opportunity to observe. *Journal of Applied Psychology*, 1990, vol. 75, no 3, pp. 322-327.
4. Viswesvaran C., Ones D. S., Schmidt F. L. Comparative analysis of reliability of job performance ratings. *Journal of Applied Psychology*, 1996, vol. 81, no 5, pp.557-574.
5. Gavriliev E. I., Avdeenko T. V. Model and Procedure for Assessing the Qualification of a Software Developer. 2022 *IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM)*, 2022, pp. 303-307.
6. Gavriliev E. I., Avdeenko T. V. Procedura ocenki kvalifikacii razrabotchika programmnogo obespecheniya [Procedure for assessing the qualifications of a software developer]. *Nauka. Tekhnologii. Innovacii: Collection of scientific papers*. In 10 parts, Novosibirsk, 06-10 December, 2021, pp. 145-148.
7. 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, no. 12, pp. 889-903.
8. Surakka S. What subjects and skills are important for software developers. *Communications of the ACM*, 2007, vol. 50, no. 1, pp. 73-78.
9. Li P. L., Ko A. J., Begel A. What Makes a Great Software Engineer? *37th International Conference on Software Engineering*, 2015, pp. 700-710.
10. 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*, 2018.
11. Bergersen G. R., Sjøberg D. I. K., Dybå T. Construction and validation of an instrument for measuring programming skill. *IEEE Trans. Softw. Eng.*, 2014, vol. 40, no. 12, pp. 1163-1184.
12. Juristo N., Moreno A.M., Stigel W. Guest editors' introduction: Software testing practices in industry. *IEEE Software*, 2006, vol. 23, no. 4, pp. 19-21.
13. Florea R., Stray V. The skills that employers look for in software testers. *Software Quality Journal*, 2019, no. 27, pp. 1449-1479.
14. Iivonen J., Mäntylä M.V., Itkonen J. Characteristics of high performing testers: a case. *ESEM '10: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 2019, pp. 1-9.
15. Deak A. What Characterizes a Good Software Tester? – A Survey in Four Norwegian Companies. *Lecture Notes in Computer Science*, 2014, no. 8763, pp. 162-172.

16. Gousios G, Kalliamvakou D., Spinellis E.D. Measuring developer contribution from software repository data. In *Proceedings of the 2008 international working conference on Mining software repositories, MSR '08*, 2008, pp. 129-132.
17. Lanubile F., Ebert C., Prikladnicki R., Herzig R.K. A Collaboration Tools for Global Software Engineering. *IEEE Software*, 2010, vol. 27, no. 2, pp. 52–55.
18. Zolkifli N. N., Ngah A., Deraman A. Version Control System: A Review. *Lecture Notes in Computer Science*, 2018, vol. 135, pp. 408-415.
19. Makashov P.A., Romanenko N.A. Servis-orientirovannyj podhod k upravleniyu IT proektami na primere ispol'zova-niya programmnogo produkta «JIRA» [Service-oriented approach to IT project management on the example of using JIRA software product]. *Sovremennye informacionnye tekhnologii i IT-obrazovanie*, 2015, vol. 11, no. 2, pp. 127–132.
20. Collins E. F., Lucena Jr. V. Software Test Automation practices in agile development environment: An industry experience report. *2012 7th International Workshop on Automation of Software Test (AST)*, 2012, pp. 57–65.
21. Apraci I., Al-Emran M., Al-Sharafi M. A. The impact of knowledge management practices on the acceptance of Massive Open Online Courses (MOOCs) by engineering students: A cross-cultural comparison. *Telematics and Informatics*, 2020, no. 54, pp. 1–13.
22. Gavriliev E. I., Avdeenko T. V. Mnogofaktornaya regressionnaya model' ocenki kvalifikacii testirovshchika programm-nogo obespecheniya [Multivariate regression model for assessing the qualifications of a software tester]. *Doklady TUSUR*, 2022, vol. 25, no. 4, pp. 115–121.
23. Tomita-parser, available at: <https://tech.yandex.ru/tomita/> (accessed: 02.04.2023).
24. Kaiser-Meyer-Olkin (KMO) Test for Sampling Adequacy, available at: <https://www.statisticshowto.com/kaiser-meyer-olkin/> (accessed: 16.04.2023).
25. Understanding Data Mapping Tools, Process, and Techniques, available at: <https://www.astera.com/type/blog/understanding-data-mapping-and-its-techniques/> (accessed: 02.04.2023).

Erchimen I. Gavriliev

Postgraduate student, Department of Theoretical and Applied Computer Science, Novosibirsk State Technical University (NSTU, Russia, 630073, Novosibirsk, Karl Marks Ave. 20), e-mail: erchimen_gavriliev@outlook.com, ORCID ID: 0000-0001-7289-3969.

Tatiana V. Avdeenko

Dr. of Sci. (Engineering), Professor, Department of Theoretical and Applied Computer Science, Novosibirsk State Technical University (NSTU, Russia, 630073, Novosibirsk, Karl Marks Ave. 20), e-mail: tavdeenko@mail.ru, ORCID ID: 0000-0002-8614-5934.