

Научная статья
УДК 004.413.4
© И. В. Кучумов
DOI: 10.24412/2225-8264-
2024-1-98-103

Ключевые слова: мониторинг, выгорание, команда, командообразование, программист, программирование, мотивация

Keywords: monitoring, burnout, team, teambuilding, programmer, programming, motivation

МОНИТОРИНГ ПРОЦЕССА ВЫГОРАНИЯ И СИСТЕМНЫЕ ОСОБЕННОСТИ МОТИВАЦИИ КОМАНДЫ ПРОГРАММИСТОВ

Кучумов И. В.¹

***Аннотация.** Актуальность проблемы создания надежных программ, индустриального и многопрофильного программирования растет в связи с ростом объемов и сложности задач автоматизации и усложнением процессов и продуктов на рынке программных продуктов. Этому способствуют и практические проблемы экономики, производства, оказания услуг, ориентированных на учет потребительских ожиданий. Цель нашего исследования заключается в исследовании такой многосторонней проблемы, как выгорание программистов, программистских групп в современных рыночных (продуктовых) и цифровых условиях. В исследовании применены методы: системный анализ-синтез, интеграция, методы принятия решений, идентификация и шкалирование, анкетирование (тестирование), ситуационное прогнозирование и моделирование, социальная инженерия. Проведен системный и сравнительный анализ целей, задач, условий, воздействий на эффект выгорания программистов и получены следующие результаты анализа: 1) причин и условий феномена выгорания программистов; 2) факторов, влияющих на выгорание с учетом неопределенностей и рисков в окружении и рассматриваемой инфраструктуре; 3) целей мониторинга выгорания и роли руководителя программистской команды в снижении процесса выгорания; 4) принципов мотивации членов команды и повышения их компетенций; 5) для последнего типа анализа разработана система тестовых вопросов (опросная анкета), которая позволит идентифицировать ключевые факторы выгорания и ранжировать их по важности в процессе. Значимость результатов исследований заключается как в теоретическом (системном) интересе к ним, так и в практической их ориентации, в частности, в возможности их использования для решения задач формирования команд программистов, ведения командной работы и мотивации программистов.*

¹Кучумов Илья Владимирович — руководитель отдела разработки, компания «Яндекс» (Россия, г. Москва, ул. Льва Толстого, 16)
E-mail: kuchumov.ilya@gmail.com
ORCID: <https://orcid.org/0009-0007-2396-1946>

MONITORING THE BURNOUT PROCESS AND SYSTEMIC FEATURES OF MOTIVATION OF A TEAM OF PROGRAMMERS

Ilya V. Kuchumov

Yandex Company, Moscow, Russian Federation

***Abstract.** The relevance of the problem of creating reliable programs, industrial and multidisciplinary programming is growing due to the growth of the volume and complexity of automation tasks and the complication of processes and products in the software market. This is also facilitated by practical problems of the economy, production, and the provision of services focused on accounting for consumer expectations. The purpose of our study is to investigate such a multi-faceted problem as the burnout of programmers, programming groups in modern market (product) and digital conditions. The study applied methods: system analysis-synthesis, integration, decision-making methods, identification and scaling, questionnaire (testing), situational forecasting and modeling, social engineering. A systematic and comparative analysis of the goals, objectives, conditions, impacts on the effect of programmers burnout was carried out and the following main results were obtained, analysis results: 1) causes and conditions of the phenomenon of programmers burnout; 2) factors affecting burnout taking into account uncertainties and risks in the environment and the infrastructure under consideration; 3) goals of monitoring burnup and the role of the head of the programming team in reducing the burnup process; 4) the principles of motivating team members and increasing their competencies; 5) for the last type of analysis, a system of test questions (survey questionnaire) has been developed, which will identify key burnout factors and rank them by importance in the process. The significance of the research results lies both in the theoretical (system) interest in them, and in their practical orientation, in particular, in the possibility of using them to solve the problems of forming teams of programmers, conducting teamwork and motivating programmers*

Поступила в редакцию:
27.01.2024

ВВЕДЕНИЕ

Оценивание сложности программного обеспечения (ПО) является многосторонней и сложной проблемой, как по динамике, так и структуре, вычислительным аспектам [1] и др.

Несмотря на автоматизацию программирования и появление некоторых интеллектуальных алгоритмов автоматизировать полностью процесс невозможно — слишком слабо формализуемы и структурируемы многочисленные нестандартные (нешаблонные) ситуации. Доказательное (верифицирующее) программирование базируется на строгих, аксиоматического уровня знаниях и редко используется. Например, для этого необходимо знать алгебраические системы (в частности, теорию полугрупп, абстрактных автоматов) [2].

Оценка качества программы ориентирована, например, на оценку времени, памяти, интерактивности, надежности и других требований по специальной программе и методике испытаний, с соблюдением требований рынка к конкурентоспособности программного продукта, в частности, совместимости, кросс-платформенности, единства интерфейса, парадигм DevOps, Agile, экстремального программирования [3, 4] и других методологии. Проверятся устойчивость, надежность и восстанавливаемость программы.

Идентификация начала процесса выгорания (BurnOut [5]) позволяет не только предпринимать соответствующие меры, но и выявлять причины, улучшать профессиональный климат и управляемость коллектива. Но этот феномен осложнен многофакторностью, неопределенностью, сложностью и слабой прогнозируемостью. Исследователи рассматривают различные причины — стрессовые, адаптационные, психосоматические, самоорганизационные и др. Но выгорание не сводимо ни к одной из этих причин, являясь их интегральным результатом.

Степень профессионального выгорания персонала исследуется различными методиками, например, методикой, представленной в исследовании О. И. Муравьевой и К. В. Козловой [6]. Есть опросники (МБИ, EPI, SPFQ и др.), оценочные критерии (Спилбергер-Ханина, Крускала-Уоллиса, Спирмена) [7].

ВОЗ определяет категорию «выгорание» как синдром непреодоленного хронического стресса, физической и психологической усталости на рабочем месте [8, 9].

МОНИТОРИНГ МОТИВАЦИИ ПРОГРАММИСТОВ И ТЕСТИРОВЩИКОВ

Программисты обычно работают с ассоциированными тестировщиками, участвуют в процессе тестирования, ведь и программист, и тестировщик ответственны за устойчивое и результативное ПО. Достаточно сложный программный комплекс проектируется и разрабатывается командой специалистов. Например, программистской группой из 5-7 человек во главе с руководителем группы.

Руководитель программистской команды соблюдает следующие принципы:

1) «разделяй работу на модули по компетенциям исполнителей»;

2) «веди контроль и координацию работ ежедневно»;

3) «указывай релевантный инструмент разработки»;

4) «держи в уме ключевые связи модулей — информационные, интерфейсные, функциональные и вести настройку всего проекта по спецификациям».

Команда может включать не только тестировщиков, но и программистов, математиков, специалистов по инженерии и аналитике данных и др. После того как команда сформирована, важно мотивировать членов команды. В связи с этим, используют семь нижеследующих принципа мотивации членов команды:

1. «Знае команды», знать команды, которые работают на успех и продуктивную гармонию, знать отвлекающие от формирования успешных команд факторы;

2. «Физическое разделение», разделять команду тестировщиков и остальную часть команды проекта, но располагая команду разработчиков и дизайнеров ближе;

3. «Быть справедливым», несправедливость проявляется по-разному (заработная плата, вознаграждение, рабочее место, сверхурочная работа и др.);

4. «Наличие взаимопонимания», хорошая коммуникация является одним из ключевых компонентов рабочих отношений, это позволяет принимать обоснованные решения;

5. «Разделение целей», команда разработчиков всегда со своим видением, четким направлением, направленным на продуктивность, с четкостью целей группы тестирования и «кодеров»;

6. «Исключение дублирования усилий», применение простых методов (обзоры, распределенное и парное тестирование, ограничение и управление временем);

7. «Усиление поддержки менеджмента», если команда считает, что руководство не поддерживает деятельность всей команды, то это может иметь серьезные негативные последствия для команды.

Применяются эффективные методы и адаптивные среды учета обратных связей, контроля и самоконтроля, перехода от внешней стимуляции к стимуляции внутренней. Важно совершенствовать функциональные обязанности специалистов при разработке сложных комплексов.

В процесс тестирования вовлекаются команды QA (Quality Assurance) для раннего выявления критических дефектов. На наш взгляд, QA-инженер должен обладать компетенциями (основами):

— индуктивной, дедуктивной и модульной методологии;

— системного анализа-синтеза (статистики и причин отказов программ);

— разработки и применения тест-кейсов;

— составления стратегии тестирования, тестового сценария;

— документирования (баг-трекинга, аналитики и др.);

— критериями оценки качества (отдельных программ, комплекса);

— информационного и функционального тестирования интерфейса;

— тестирования в среде API;
— языков HTML, CSS и JS (возможно, C++, SQL, Python, Ruby-on-Rails и др.);

— методологий современной и гибкой разработки системы тестов и сценария тестирования (SoftSkills, Agile, Scrum, Kanban, Scrumban, DevTools, API и др.).

Компетенции можно протестировать с помощью опросных анкет. Разработана, например, анкета с вопросами, ее фрагмент приводится ниже.

1. Повышение компетенций сдерживается:

- а) квалификацией в области программирования (тестирования)?
- б) непониманием задач?
- в) отсутствием личной мотивации?
- г) отсутствием качественных тренингов, курсов?
- д) слабостью информационной инфраструктуры создания программ?
- е) недостаточной рыночной востребованностью?
- ж) отсутствием осязаемого результата, рутинной;
- з) отсутствием отдыха («перегруз» и «срочность») и интереса к работе.

2. Расположить по мере актуальности следующие способы:

- а) веб-моделирование;
- б) командная, проектная работа;
- в) веб-тестирование (интернет, онлайн);
- г) визуализация состояния (связей) продукта.

3. В обязанности тестировщика должны входить:

- а) тестирование программы как «черный ящик»;
 - б) «баг-трекинг» (bug-tracking) или отслеживание ошибок;
 - в) тестирование функционала;
 - г) тестирование интерфейса;
 - д) тестирование взаимодействий с базой данных;
 - е) документирование тестирования;
 - ж) знать английский (хотя бы «Immediate+»);
 - з) внесение корректирующих изменений в код.
- В качестве возможных ответов предлагаются:
- А) «определенно — нет», уверенность на 95-100%;
 - Б) «нет», уверенность на 75-94%;
 - В) «затрудняюсь ответить», уверенность на 45-55% (нейтральность);
 - Г) «скорее — да», уверенность на 75-94%;
 - Д) «определенно — да», уверенность на 95-100%.

Анализ процесса выгорания (BurnOut Process) позволит выявить причины плохой управляемости команды. Профессиональное выгорание команды является предметом исследований, соответствующих опросов многих узких специалистов. Управление командой опирается на культуру, мотивацию, раскрытие компетенций и снятие усталости специалиста.

Программирование (тестирование) — сложный, плохо структурируемый процесс, с выгоранием профессионала. Велика роль лидера, руководителя команды при реализации, например, обратных связей в группе (между группами), командного духа и раскрытия способностей каждого разработчика и тестировщика, ситуационного лидерства.

Профессионалов экспертного уровня мало, следует привлекать методы моделирования, прогнозирования, интеллектуального принятия решений.

ВЫГОРАНИЕ ПРОГРАММИСТОВ КАК СИСТЕМНЫЙ ПРОЦЕСС

Процесс выгорания программистов проявляется в снижении стремления к новым компетенциям, в забывании, откладывании «в долгий ящик», психолого-физиологической усталости, влияющей на логико-алгоритмическое мышление. Падает и эффективность желания программистов учить друг друга, инициативность и осознанная ответственность, обучаемость.

Ведущий или главный программист группы (обычно небольшой, 5-7 человек) — это лидер, профессионал высокого уровня, обладатель высокого эмоционального интеллекта, приверженец командного стиля поведения и оценки программиста «по вкладу». Стиль его руководства направлен на развитие, саморазвитие команды. Лидер группы, имеющий эффективную связь с командой, может определить дисбаланс в команде и процессах, нивелировать риски стресса.

Отметим основные составляющие управления программистской группой, создания и поддержки единого поля проекта, разработки:

- 1) согласование общегрупповых целей, спецификаций и структуры целевых действий;
- 2) поддержка активной и результативной обратной связи в группе (между группами) и инновационных технологий, в том числе, групповой динамики;
- 3) ориентация на обеспеченный доступными ресурсами целевой и оперативный выбор решений, их оценивание и улучшение качества;
- 4) обеспечение партнерства и командного духа в группе, раскрытие творческих способностей каждого программиста;
- 5) коучинговый, консалтинговый и компетентностный подход к каждому программисту и каждой ситуации (принцип ситуационного лидерства).

Полезно использовать различные, адаптируемые к группе и проекту модели руководства. Например, модель Херси-Бланшара [10] ситуационного руководства использует стили управления:

- командный — для недостаточно профессиональных программистов-энтузиастов;
- наставнический — для программистов с определенными компетенциями, но демотивированных, например, из-за неудач;
- поддерживающий — для компетентных и недостаточно внутренне мотивированных;
- делегирующий — для высокомотивированных профессионалов.

Программирование — искусство «пересказа» компьютеру замысла алгоритма. Программированием занимаются творческие люди, для которых часто целью является не заработок (не только он), а профессиональный интерес, творческое начало проекта и желание «умственного состязания», как вида страсти. Последнее обстоятельство сильно влияет на выгорание — «невыгорательный» проект должен быть продуктивным, ценным.

Сама профессия не только тяготеет к искусству и имеет многогранный характер, но и предъявляет повышенные требования к «носителям» компетенций. Необходимо непрерывное самосовершенствование, освоение языков и технологий программирования, на-

пример, процедурного, функционального, экстремального, потокового, «на лету» и др. Требуются знания и применение релевантного формально-верификационного аппарата, коммуникативные компетенции.

Работа в команде, с важным проектом, взаимодействия с заказчиками и стейкхолдерами становятся источниками эмоциональной напряженности для профессионала, его высокой концентрации, прогнозных и эвристических способностей. Личные достижения члена команды программистов вносят большой вклад в их выгорание, эмоциональное истощение или деперсонализацию. Это объясняется, скорее всего, эффективностью вклада личности, поддержкой высокой самооценки и заниженной оценкой личных профессиональных возможностей. Влияет и чувствительность к признанию, чувство незаменимости, принятие на себя непосильных обязательств.

Удерживать талантливых программистов сложно и «накладно», влияет:

- 1) конкуренция других команд;
- 2) давление «цифровизации», необходимость обучения новым подходам;
- 3) когнитивная нагрузка на программистов и группу;
- 4) сложность интеграции и управления ею, например, облачными и туманными сервисами без специальной ИТ-подготовки;
- 5) сложность автоматизации разработки и тестирования ПО, работы с данными, например, с помощью Big Data, ЦОД и др.

Есть внешние и внутренние причины и факторы выгорания программистов. К внутренним причинам относят, например, идеалистическое ожидание с подавлением своих потребностей, чувство незаменимости с переоценкой потенциала, «размывание» границ офиса и дома. К внешним причинам — требовательность и негибкость руководителя, нечеткость целей и ресурсов, слабая мотивация (не обязательно материальная), дискомфортная рабочая среда и др.

Команда программистов, взаимодействующих с единой целью, едиными ресурсами работает по принципу сотрудничества, единства компетенций и возможностей. Она нужна для достижения целей, поэтому нужны компетенции, умения и информационно-логическая поддержка. Если один программист не потянет проект, то команда сможет. Если программисту-фрилансеру работать одному лучше, то в условиях цифровых трансформаций выигрывает команда, которая может генерировать и исследовать сложные и трудоемкие, удачные идеи и решения.

В команде программистов необходимо всем быть профессионалами, лишь тогда командный эффект будет значительным. Профессионализм — это умение слушать, отвечать, излагать, спорить без нарушения личного пространства каждого, не обижая и налаживая коммуникации и обратные связи с каждым. Тогда команда «функциональна», а ее эффективность складывается при формировании, которое происходит по принципу объединения всех для достижения цели.

При формировании команды программистов необ-

ходимо пройти все этапы, «стартовать» проект. Далее актуальны, например, ответы на вопросы:

- каковы компетенции, необходимые для реализации проекта?
- какие нужны взаимодействия для этого?
- насколько они взаимозаменяемы?

Ответы используются для выявления мотивации членов команды. После этого приступаем к тимбилдингу, набору ситуационных и игровых сценариев, психологических тестов, направленных на сплочение коллектива.

Тимбилдинг в программистской среде предполагает:

- 1) взаимодополняющие компетенции каждого, логико-аналитические способности каждого, идентификацию слабых и сильных сторон каждого (отметим, это невозможно сделать в повседневной «рутине»);
- 2) повышение сотрудничества, доверия и взаимопонимания, осознания, что в коллективе более выгодно, чем при конкуренции работников, что можно эффективно достичь целей сплоченной командой;
- 3) независимость оценок ключевых компетенций программистов команды, когда все их возможности проявляются на практике;
- 4) синхронизацию действий каждого с командными целями (поэтому популярными остаются методы «мозгового штурма», Дельфи).

Если действия в группе — асинхронные, т.е. происходят в разных направлениях, со своими целями, то командообразование формирует явление «лайнмен» или движение к цели с синхронизацией личных и командных устремлений. Менеджеру облегчается внедрение командных организационных изменений и управление ими, например, руководитель группы отвечает за доведение до каждого программиста целей проекта.

Agile-подход, сотрудничество лидеров программистских групп при итерационном наращивании ценности ПО позволит добиться:

- 1) организационной культуры и гибкости;
- 2) вовлеченности потребителя, учета его потребительских устремлений;
- 3) рациональности, точности, мобильности и оперативности;
- 4) когнитивности, интеллектуальности и системности управления;
- 5) эмоционально-психологической общности исполнителей в группе и др.

Программная системная инженерия и ее системный инструментарий адаптируют технологические и методологические подходы и ресурсы к потребительским ожиданиям. В этом направлении применяются различные системные (методологические) подходы. Например, Agile-подход включает разнообразные методы, в частности, FDD или разработки с управляемой функциональностью, экстремального программирования, Scrum или демонстрации заказчику реализованной части проекта, прототипа, макета.

Ключевой моделью бережливого лидерства для программного проекта может стать, например, модель «Дао» — на основе высоких требований руководителя и уважения к программисту команды. Использование

KPI-оценок эффективности работы в программистской группе или веб-студии [11] потребует мотивации, совершенствования обратных связей, мониторинга и внедрения методов бережливого производства (Agile, Scrum, Lean) программного продукта.

Успешность (вознаграждение) члена команды и лидера определяется результативностью, например, способностью экстренно вмешиваться, ускорять достижение результата согласно «дереву целей». Для программистских групп можно использовать SAFe — фреймворк от Scaled Agile для гибкой и слаженной разработки комплекса программ [12], не выходя за бюджет проекта.

Методы выявления особенностей выгорания программистов — различны по подходам и учитываемым аспектам. Часто рассматриваются статистические методы (ранговой корреляции Спирмена, U-критерия Манна-Уитни [13] и др.) и эмоционально-когнитивные особенности, приводящие к выгоранию в процессе работы. К таким особенностям относятся, например, эмоциональная напряженность, психологическая неустойчивость, чувство потери когнитивного контроля над ресурсами и процессами.

Эффективным методом (технологией) выявления выгорания стал метод айтрекинга, нейро-айтрекинга или отслеживания состояния по быстрым перемещениям глаз, расширению зрачка, частоте и продолжительности зрительного внимания на экране (программе) и партнере [14]. Анализируя «движение взгляда» от глазного яблока до плоскости экрана (объекта) можно понять закономерности мыслительного мозгового процесса с учетом нейрофизиологических и физических законов, например, айтрекер SMI RED-m и аналитический пакет BeGaze-SMI [15] могут построить «конус взгляда» по положению зрачка, отблескам и обозреваемому конусу точек.

Алгоритмов айтрекинга много, например, реализация процедуры Виолы-Джонса [16], с поиском матриц представления объекта x (средневзвешенных значений признаков):

$$F(x, y; t) = \sum_{i=0}^x \sum_{j=0}^y f(i, j; t)$$

где $f(i, j; t)$ — сигнал, отражение текущего состояния $(i, j; t)$ в момент t .

Можно использовать в качестве идентификационного критерия (меры близости $\varepsilon > 0$) и метрику евклидова пространства

$$\rho(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2},$$

x — объект класса, y — объекты, попадающие в класс (таксоне) по ε -критерию вида: $\rho(X, Y) < \varepsilon$.

Процедура позволяет классифицировать тип выгорания программиста, построить рейтинг мер, препятствующих выгоранию и использовать их на практике.

ЗАКЛЮЧЕНИЕ

Незнание (непонимание) последствий и вреда от вредных привычек тормозит развитие программистов, групп, их отклики на часто противоречивые сигналы менеджера, стейкхолдера. Релевантно не учитываются метрики, независимость от фреймворка команды, например, команда много работает — «бюджета больше съедает». Карьерный рост также снижается из-за несоблюдения нижеследующих практических установок для программистских групп, индивидуальной работы каждого программиста.

Выгорание программиста — частый продукт плохой постановки целей и планирования ресурсов, отсутствия продуманной мотивации, поддержки и критериев оценки результативности. Проблема выгорания и айтрекинга, распознавания и отслеживания — задача актуальная технологически и организационно, эффективно решаемая и без сложного обеспечения, инвентария.

Исследования развиваемы, в частности, можно акцентировать метод нейро-айтрекинга, применить рейтингование программистов (команд) по выгоранию. Есть перспективы и в направлении применения гештальт-психологии.

Список источников

1. Поначугин А. В. Определение надежности программного обеспечения в структуре современной информационной системы // Кибернетика и программирование. 2019. №2. С. 65-72. DOI:10.25136/2306-4196.2019.2.20341.
2. Кучумов И. В. Тестирование программ и математическая модель поиска ошибок в программном комплексе // Известия Кабардино-Балкарского научного центра РАН. 2023. №6 (116). С.74-82. DOI: 10.35330/1991-6639-2023-6-116-74-82.
3. Гильванов Р. Г., Забродин А. В. Надежность информационных систем: учебное пособие. СПб.: ПГУПС, 2022. 85 с. ISBN 978-5-7641-1821-5.
4. Максимов М. И., Горина Е. А. Agile-методология как драйвер эффективной корпоративной культуры // Региональная и отраслевая экономика. 2023. №1. С.102-111. DOI:10.47576/2949-1916_2023_1_102.
5. Klatta M., Westrickb A., Bawaa R., Gabrama O., Blakeb A., Emersonc B. Sustained resiliency building and burnout reduction for healthcare professionals via organizational sponsored mindfulness programming // EXPLORE. 2022. Vol.18. Is.2. P. 179-186. DOI: 10.1016/j.explore.2021.04.004.
6. Муравьева О. И., Козлова К. В. Профессиональное выгорание программистов: специфичность феномена // Сибирский психологический журнал. 2019. №73. С. 98-110. DOI: 10.17223/17267080/73/6.
7. Водопьянова Н. Е., Старченкова Е. С., Наследов А. Д. Стандартизированный опросник «Профессиональное выгорание» для специалистов социономических профессий // Вестник СПбГУ (сер.12). — 2013. Вып. 4. С.17-26.
8. Польская Н. А., Мухаметзянова М. Н. Особенности эмоциональной регуляции в связи со стрессом и выгоранием // Психологические исследования. 2018. Т.11. №61. С.10-21. DOI: 10.54359/ps.v11i61.264.

9. Ярошенко Е. И. Социально-психологическая модель эмоционального выгорания личности [Электр. журнал] // Ученые записки Курского государственного ун-та. 2019. №1 (49). С.1-7.
10. Rodić M., Marić S. Leadership style and employee readiness: basic factors of leadership efficiency // *Strategic management*. 2021. Vol.26. No.1. P. 53-65 DOI: 10.5937/StraMan2101053R.
11. Kaziev V., Tyutrin N., Khizbullin F., Takhumova V., Medvedeva L. Improvement and modeling of the company's activity based on the innovative KPI system // *Journal of Fundamental and Applied Science*. 2018. Vol. 10 (5S). Pp. 1406-1415. DOI: 10.4314/jfas.v10i5s.117.
12. Степанов Д. Ю., Вельсовский А. В. Применение Agile Scrum в проектах SAP // *Корпоративные информационные системы*. 2018. №1. С.9-17.
13. Бестужев-Лада И. В. Рабочая книга по прогнозированию. М.: «Мысль». 1982. 426 с.
14. Ashby N., Johnson J., Krajbich I., Wedel M. Applications and innovations of eye-movement research in judgment and decision making // *Journal of Behavioral Decision Making*. 2016. No.29 (2-3). Pp.96-102.
15. Соловьева В. А., Вениг С. Б., Белых Т. В. Анализ окулomotorной активности, наблюдаемой при изучении образовательного материала с экрана // *Интеграция образования*. 2021. Т.25. №1. С. 91-109. DOI: 10.15507/1991-9468.102.025.202101.091-109.
16. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // *IEEE Conf. on Computer Vision and Pattern Recognition*. Kauai. Hawaii. USA. 2001. Vol. 1. P. 511-518.

References

1. Panachugin A. V. Determining the reliability of software in the structure of a modern information system. *Kibernetika i Programirovanie = Cybernetics and Programming*. 2019; 2: 65-72. (In Russ.). DOI:10.25136/2306-4196.2019.2.20341.
2. Kuchumov I. V. Testing programs and a mathematical model for finding errors in the software complex. *Izvesiya Kabardino-Balkarskogo Nauchnogo Tcentra RAN = Izvesiya Kabardino-Balkarian Scientific Center RAS*. 2023; 6: 74-82. (In Russ). DOI: 10.35330/1991-6639-2023-6-116-74-82.
3. Gil'vanov R. G., Zabrodin A. V. Reliability of information systems: tutorial. St. Petersburg: PGUPS. 2022. Pp. 1-85.
4. Maksimov M. I., Gorina E. A. Agile-methodology as a driver of an effective corporate culture. *Regional'naya i otraslevaya ekonomika = Regional and Industry Economy*. 2023; 1: 102-111. (In Russ). DOI:10.47576/2949-1916_2023_1_102.
5. Klatta M., Westrickb A., Bawaa R., Gabrama O., Blakeb A., Emersonc B. Sustained resiliency building and burnout reduction for healthcare professionals via organizational sponsored mindfulness programming. *EXPLORE* 2022; (18, is. 2); 179-186. DOI: 10.1016/j.explore.2021.04.004.
6. Murav'eva O. I., Kozlova K. V. Professional burnout of programmers: specificity of the phenomenon. *Sibirskii psihologicheskii gurnal = Siberian Psychological Journal*. 2019; 73: 98-110. (In Russ). DOI: 10.17223/17267080/73/6.
7. Vodopj'anova N. E., Starchenkova E. S., Nasledov A. D. Standardized questionnaire «Professional burnout» for specialists in socioeconomic professions. *Vestnik Sankt-Peterburgskogo universiteta = Bulletin Sankt-Peterburg University* (ser.12), 2013; 4: 17-26. (In Russ).
8. Pol'skaya N. A., Muhametdzjanova M. N. Features of emotional regulation in connection with stress and burnout. *Psihologicheskii issledovaniya = Psychological research*. 2018; 61: 10-21. (In Russ). DOI: 10.54359/ps.v11i61.264.
9. Yaroshenko E. I. Socio-psychological model of emotional burnout of personality. [Electr. Journal]. *Ucennie zapiski Kurskogo gosudarstvennogo universiteta = Academic notes of Kursk State University*. 2019; 1: 1-7. (In Russ).
10. Rodić M., Marić S. Leadership style and employee readiness: basic factors of leadership efficiency. *Strategic management*. 2021; 26: 53-65 DOI: 10.5937/StraMan2101053R.
11. Kaziev V., Tyutrin N., Khizbullin F., Takhumova V., Medvedeva L. Improvement and modeling of the company's activity based on the innovative KPI system. *Journal of Fundamental and Applied Science*. 2018; 10 (5S): 1406-1415. (In Russ). DOI: 10.4314/jfas.v10i5s.117.
12. Stepanov D. Yu., Vel'sovskii A. V. Application of Agile Scrum in SAP Projects. *Korporativnie informatsionnie sistemi = Corporate Information Systems*. 2018; 1: 9-17. (In Russ).
13. Bestugev-Lada I. V. Forecasting Workbook. Moscow, 1982. 426 p.
14. Ashby N., Johnson J., Krajbich I., Wedel M. Applications and innovations of eye-movement research in judgment and decision making. *Journal of Behavioral Decision Making*. 2016; 29: 96-102.
15. Solov'eva V. A., Venig C. B., Belih T. V. Analysis of oculomotor activity observed when studying educational material from the screen. *Integratsiya obrazovaniya = Integration of education*. 2021; 25 (1): 91-109. (In Russ). DOI: 10.15507/1991-9468.102.025.202101.091-109.
16. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. *IEEE Conf. on Computer Vision and Pattern Recognition*. Kauai. Hawaii. USA; 2001; 1: 511-518.