

Выявление рисков системы контроля технологических операций ремонта электродвигателей по методике АСС

П. А. Микишева, Н. И. Мустафина*

ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики»,
г. Пермь, Российская Федерация

Адрес: 614070, Российская Федерация, г. Пермь, ул. Студенческая, д. 38

* nazgul-2003@mail.ru

Аннотация

В процессе обучения по образовательной программе «Программная инженерия» в Пермском филиале НИУ ВШЭ управлению рисками студенты проходят методику АСС. С помощью данной методики была проанализирована система контроля технологических операций ремонта электродвигателей. Задачи системы – обеспечить назначение операций для ремонта пришедшему в компанию двигателю, сохранить время начала и окончания операций, хранить информацию о работниках, ответственных за каждую операцию и позволить просматривать списки законченных и длящихся ремонтов. Методика АСС была выбрана для анализа системы так как она позволяет найти скрытые проблемные места приложения и оценить вероятность их поломки. Применение АСС должно помочь увеличить качество системы. Суть применения АСС – анализ системы по нескольким параметрам: выделение атрибутов (прилагательные – цели и назначение продукта), компонентов (существительные – части системы) и акторов (пользователи), на пересечении параметров стоят возможности системы (глаголы – что продукт может делать), для которых оценивается их величина риска по двум измерениям – частота и критичность. Потом была составлена единая таблица, где строки – это атрибуты, столбцы – компоненты, а на их пересечении стоят возможности. Каждой возможности приписывается её величина риска и составляется общая тепловая карта, где красный – наиболее сильные риски, зелёный наименее сильные, что позволило тестировать систему эффективнее, так как в начале тестируются возможности с наибольшей величиной риска. По итогам проделанной работы, был составлен тест-план, тест-кейсы и проведено сценарное тестирование, что позволило увеличить качество системы. После применения АСС была проведена переоценка системы и оценена результативность АСС в сравнении с другими методиками.

Ключевые слова: тестирование, реестр рисков, тест-план, тест-кейсы, система контроля технологических операций, ремонт электродвигателей, методика АСС, выявление рисков, технологические операции, контроль качества, производственные процессы, прогнозирование сбоев, улучшение эффективности, техника безопасности, оценка рисков, повышение производительности, оптимизация процессов, профилактика поломок, мониторинг процесса ремонта, анализ данных, программная инженерия, учебный процесс, университет

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Микишева П. А., Мустафина Н. И. Выявление рисков системы контроля технологических операций ремонта электродвигателей по методике АСС // Современные информационные технологии и ИТ-образование. 2024. Т. 20, № 1. С. 164-174. <https://doi.org/10.25559/SITITO.020.202401.164-174>

© Микишева П. А., Мустафина Н. И., 2024



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Attitude of Students of Computer Science Specialites to Including Computer Simulators and Business Games in the Academic Process

P. A. Mikisheva, N. I. Mustafina*

National Research University Higher School of Economics, Perm, Russian Federation

Address: 38 Studencheskaya St., Perm 614070, Russian Federation

* nazgul-2003@mail.ru

Abstract

In the process of training in the Perm branch of the Higher School of Economics, risk management students pass the ACC methodology. With the help of this methodology the system of control of technological operations of electric motor repair was analyzed. The tasks of the system are to ensure assignment of repair operations to a motor that has come to the company, to store the start and end times of operations, to store information about the employees responsible for each operation and to allow viewing the lists of completed and ongoing repairs. The ACC technique was chosen to analyze the system because it allows us to find hidden problem areas of the application and assess the probability of their failure. The application of ACC should help to increase the quality of the system. The essence of ACC application is to analyze the system according to several parameters: attributes (adjectives - goals and purpose of the product), components (nouns - parts of the system) and actors (users), at the intersection of the parameters there are system capabilities (verbs - what the product can do), for which their risk value is assessed according to two dimensions - frequency and criticality. Then a unified table was created, where rows are attributes, columns are components, and at their intersection are capabilities. Each opportunity is assigned its risk value and a general heat map is drawn up, where red are the strongest risks, green is the least strong, which allows testing the system more efficiently, since the opportunities with the highest risk value are tested first. As a result of the work done, a test plan, test cases and scenario testing was done, which increased the quality of the system. After the application of ACC, the system was reevaluated and the performance of ACC compared to other techniques was evaluated.

Keywords: testing, risk register, test plan, test cases, process operations control system, motor repair, ACC methodology, risk identification, process operations, quality control, manufacturing processes, failure prediction, efficiency improvement, safety engineering, risk assessment, productivity improvement, process optimization, failure prevention, repair process monitoring, data analysis, software engineering, training process, university

Conflict of interests: The authors declare no conflict of interest.

For citation: Mikisheva P.A., Mustafina N.I. Attitude of Students of Computer Science Specialites to Including Computer Simulators and Business Games in the Academic Process. *Modern Information Technologies and IT-Education*. 2024;20(1):164-174. <https://doi.org/10.25559/SITI-TO.020.202401.164-174>



Введение

Тестирование программного обеспечения имеет ключевое значение в процессе разработки, внедрения и сопровождения программного продукта. Его основная цель – обнаружение ошибок, дефектов и недостатков в программе, а также проверка соответствия программного продукта требованиям и ожиданиям пользователей. Существует философия, что ошибки присутствуют всегда – в любой программе [1-15]. Найти все невозможно, но если не удалось выявить ни одной – работа тестировщика провалена, поэтому необходимо тщательно проанализировать систему на наличие ошибкоопасных мест. Тестирование не должно быть самоцелью, так как создание идеального тестового плана зачастую неэффективно и занимает много времени. Хороший план создается в начале разработки и может изменяться по мере развития проекта. В больших проектах могут возникать проблемы с созданием наборов тестов без общей структуры, что затрудняет их объединение. Компания Google также столкнулась с этими проблемами и разработала методику АСС-анализа [16]. Существуют разные методики построения тест-планов, такие как IEEE и RUP, однако они не касаются содержания тестирования, в них не описано, какие части системы и каким образом проверять. IEEE фокусируется на определении целей и задач тестирования, разработке стратегии тестирования и планировании процесса. RUP же акцентирует внимание на создании тестовых сценариев, выполнении тестирования и анализе результатов. А методика АСС (Attribute, Component, Capability) предлагает систематический подход к анализу системы и разработке тест-плана. Она помогает выявить потенциальные проблемы и риски, связанные с функционированием системы, и разработать план действий для их устранения. В рамках методики осуществляется анализ компонентов системы, их атрибутов и возможностей, что позволяет рассмотреть систему с точки зрения пользователей и выявить места, в которых могут возникнуть ошибки или проблемы.

С учетом эффективности данного метода она была выбрана для выявления рисков системы контроля технологических операций ремонта электродвигателей. В рамках данной системы осуществляется процесс сопровождения неисправного двигателя, что предполагает назначение операций двигателю, работникам, ответственных за него, а также просмотр состояния завершения ремонта. Для анализа системы по методике АСС были выделены ее компоненты, атрибуты и возможности, а также акторы – роли пользователей системы. Была составлена тепловая карта системы, на которой были отмечены потенциальные риски. Далее были назначены частотность и критичность для каждого риска, составлены реестр рисков и тест-план и тест-кейсы на их основе для дальнейшего тестирования системы.

Этот метод позволяет систематизировать процесс анализа системы, выявить ошибкоопасные места и повысить ее надежность и эффективность, а также улучшить качество работы.

В настоящей статье приводятся результаты анализа системы контроля технологических операций ремонта электродвигателей, проведенный по методике АСС. Цель исследования – выявить риски системы с помощью методики АСС и улучшить ее качество.

Материалы и методы

В качестве материалов использовалась система контроля технологических операций ремонта электродвигателей. Описание работы системы:

1. В фирму приходит на ремонт сломанный двигатель, он становится на учёт – ему назначают номер и список работ;
2. Карточка ремонта перемещается в список “В работе”, где список незаконченных ремонтов виден всем пользователям;
3. Начальник цеха назначает мастера, ответственного за каждую операцию;
4. Мастер назначает конкретных рабочих на операцию, он же отвечает за начало и конец выполнения работ;
5. Карточки двигателей автоматически перемещаются из списка “В работе” в список “Выполненные” после того как ремонт закончен;
6. Директор может сформировать для себя отчеты, представляющие информацию, какой рабочий и когда выполнял конкретную операцию;
7. Администратор может выполнять операции добавления, редактирования и удаления для пользователей. И назначать им роли;
8. Также администратор может выполнять операции по добавлению, редактированию и удалению групп работ, операций и заказчиков.

Для анализа была применена методика АСС. Методика АСС предназначена для эффективного составления плана тестирования. Главные вопросы, на которые отвечает АСС анализ – это как распределить ресурсы, определённые на тестирование между различными функциями и компонентами системы и как найти критерий завершенности тестирования. Продукт в АСС анализе описывается с нескольких сторон, во-первых нужно найти качества, которые имеет продукт с точки зрения пользователя, ответить на вопрос “Какой наш продукт?” (это атрибуты системы); во-вторых нужно найти “кирпичи”, из которых система устроена, ответить на вопрос “Что есть у продукта для проявления выявленных качеств?” (это компоненты системы); в-третьих нужно понять какие возможности предоставляет продукт пользователю, ответить на вопрос “Какие действия пользователь может совершать с помощью продукта?” (это возможности системы). После этого строится таблица “компоненты – атрибуты – возможности”, где компоненты – это строки, атрибуты – столбцы, а возможности находятся на пересечении строк и столбцов, но стоит отметить, что они также могут отсутствовать. Все возможности имеют разное значение, определяемое по частоте сбоев и критичности отказа, которые измеряются в диапазоне от 1 до 4. Каждая возможность получает собственную оценку риска по произведению критичности на частоту отказа. После оценки величины риска составляется тепловая карта, ячейки в которой раскрашиваются по величине риска согласно цветам из таблицы 1. Каждая строка и столбец являют собой логическую целостность, реализующуюся при тестировании. Самое большое число ресурсов тестирования тратится на столбцы или строки, которые имеют наибольшее количество возможностей, близким к красному цвету, тем самым с самой большой суммой величин рисков.



Исследование и обсуждение

В ходе выполнения АСС анализа были выявлены такие компоненты, как: «Поиск» (поисковые строки на разных вкладках приложения), «Карта ремонта» (карточка с информацией о ремонте), «В работе» (список карт активных и приостановленных ремонтов), «Выполненные» (список карт завершённых ремонтов), «Отчёты» (вкладка для просмотра часов рабочих и информации по заказам), «Пользователи» (вкладка администрирования данных пользователей), «Группы» (вкладка администрирования групп работ), «Операции» (вкладка администрирования возможных операций с двигателем), «Заказчики» (вкладка администрирования данных о заказчиках работ); атрибуты: «Простой» (предлагает пользователю совершать только интуитивно-понятные действия), «Удобный» (минимизирует время выполнения операций для часто выполняемых действий), «Доступный» (позволяет подключаться пользователям с разными ролями), «Безопасный» (защищает информацию от сторонних угроз). При применении методики АСС в её классическом варианте предполагается ответить

только на вопросы «Какая наша система?», «Что она может дать пользователю?» и «Какие возможности она предоставляет?», однако при качественном анализе многопользовательской системы возникает ещё один вопрос «Кто будет пользоваться системой?». Поэтому нами было предложено анализировать систему с четырёх сторон, добавив АСС анализу ещё одно измерение – акторов, которые в нашем случае являются пользователями системы. Модифицированный метод АСС анализа был назван нами многомерным АСС. В нашем случае пришлось добавить всего одно дополнительное измерение, но возможно добавление и большего числа измерений для проведения еще более качественного анализа характеристик системы. Нами были выделены акторы – «Администратор», «Мастер», «Начальник цеха», «Директор», «Рабочий». После были выявлены 42 возможности, они получили своё место в общей таблице в пересечениях компонентов, атрибутов и пользователей табл. 2. Для определения величины риска каждая возможность оценивалась по факторам: частота сбоев и степень воздействия/критичность отказа табл. 1.

Таблица 1. Частота и критичность отказа
Table 1. Frequency and criticality of failure

Частота/Критичность	1. Минимальный	2. Небольшой	3. Существенный	4. Максимальный
1. Очень редко	1	2	3	4
2. Редко	2	4	6	8
3. Иногда	3	6	9	12
4. Часто	4	8	12	16

Источник: здесь и далее в статье все таблицы и рисунки составлены авторами.

Source: Hereinafter in this article all tables and figures were made by the authors.

Для визуализации величины риска была составлена тепловая карта, которая представлена в табл. 2-3, где по столбцам расположены атрибуты, а по строкам компоненты, а на их пересечении возможности по акторам.

чении возможности по акторам.

Таблица 2. Тепловая карта для атрибутов простой и удобный
Table 2. Heatmap for attributes simple and convenient

	1/ Простой: интуитивно-понятные действия		2/ Удобный: минимизация операций для часто выполняемых действий	
A/ Поиск	Администратор: 1. Искать работников по имени и электронной почте (частота - очень редко, критичность минимальная) Риск 1	Администратор: 2. Искать по ремонтному номеру конкретный заказ (частота - очень редко, критичность минимальная) Риск 1	Администратор: 1. На каждой вкладке можно искать (работы, группы, операции, отчёты, заказчики) по ключевой информации (частота - очень редко, критичность минимальная) Риск 1	
B/ Карта ремонта	Мастер: 1. Добавлять описание заказа (частота - очень редко, критичность небольшая) Риск 2	Мастер: 2. Назначать рабочих для заказа (частота - часто, критичность существенная) Риск 12	1. Заказ автоматически переходит из списка выполненных в список завершённых (частота - очень редко, критичность существенная) Риск 4	Мастер: 2. Начинать/ заканчивать/ приостанавливать ремонт (частота - часто, критичность максимальная) Риск 16



	1/ Простой: интуитивно-понятные действия			2/ Удобный: минимизация операций для часто выполняемых действий	
С/В работе	Начальник цеха: 1. Просматривать карточки не выполненных работ (частота - часто, критичность максимальная) Риск 16			Все: 1. Отображать выбранное количество текущих работ (частота - очень редко, критичность минимальна) Риск 1	
D/ Вы-полнен-ные	Начальник цеха: 1. Просматривать карточки выполненных работ (частота - часто, критичность максимальная) Риск 16			Мастер: 1. Изменить статус работы с завершенной на незавершенную (частота - иногда, критичность максимальная) Риск 12	Директор: 2. Посылать на почту уведомление о завершении работы (частота - редко, критичность минимальная) Риск 2
E/ Отче-ты	Директор: 1. Скачивать отчет (частота - редко, критичность небольшая) Риск 4			Директор: 1. Просмотреть сколько рабочий затратил время на выполнение операции (частота - иногда, критичность небольшая) Риск 6	Директор: 2. Просматривать какие операции выполняли конкретные рабочие в течение заданного периода (частота - иногда, критичность небольшая) Риск 6
F/ Пользователи	Администратор: 1. Добавлять пользователей (частота - очень редко, критичность существенная) Риск 3	Администратор: 2. Удалять пользователей (частота - очень редко, критичность существенная) Риск 3	Администратор: 3. Редактировать пользователей (частота - очень редко, критичность существенная) Риск 3	Администратор: 1. Просматривать выбранное количество записей о пользователях (частота - очень редко, критичность небольшая) Риск 2	Администратор: 2. Увольнять пользователей (частота - очень редко, критичность небольшая) Риск 2
G/ Группы	Администратор: 1. Добавлять группы (частота - очень редко, критичность существенная) Риск 3	Администратор: 2. Удалять группы (частота - очень редко, критичность существенная) Риск 3	Администратор: 3. Редактировать группы (частота - очень редко, критичность существенная) Риск 3	Администратор: 1. Добавлять операции в группу из заранее сформированного списка (частота - редко, критичность существенная) Риск 6	
H/ Операции	Администратор: 1. Добавлять операции (частота - очень редко, критичность существенная) Риск 3	Администратор: 2. Удалять операции (частота - очень редко, критичность существенная) Риск 3	Администратор: 3. Редактировать операции (частота - очень редко, критичность существенная) Риск 3		
I/ Заказчики	Администратор: 1. Добавлять заказчиков (частота - очень редко, критичность минимальная) Риск 1	Администратор: 2. Удалять заказчиков (частота - очень редко, критичность минимальная) Риск 1	Администратор: 3. Редактировать заказчиков (частота - очень редко, критичность минимальная) Риск 1	1. Автоматически отправляются сообщения на почту по завершенным заданиям (частота - редко, критичность существенная) Риск 6	



Таблица 3. Тепловая карта для атрибутов доступный и безопасный
Table 3. Heatmap for attributes accessible and secure

	3/ Доступный : позволяет подключаться пользователям с разными ролями	4/ Безопасный : защищает информацию от различных угроз
A/ Поиск	Все: 1. Искать по номеру работы и заказчику в работе и в выполненных среди текущих и завершённых работ (частота - очень редко, критичность минимальная) Риск 1	
B/ Карта ремонта	Все: 1. Просматривать статус завершенности ремонта (частота - очень редко, критичность максимальная) Риск 4	
C/ В работе	Все: 1. Просматривать все работы в активном и приостановленном состоянии (частота - редко, критичность максимальная) Риск 8	
D/ Выполненные	Все: 1. Просматривать список завершенных работ, их дату окончания, заказчика, ремонтный номер (частота - очень редко, критичность минимальная) Риск 1	Администратор: 1. Нельзя изменить мастера и работников после начала работы, что позволяет предупредить накладку в расписании рабочих (частота - очень редко, критичность существенная) Риск 3
E/ Отчеты		Директор: 1. Сохранять отчёты конфиденциальными (частота - очень редко, критичность максимальная) Риск 4
F/ Пользователи		Администратор: 1. Назначать роли с ограниченными правами доступа (частота - редко, критичность максимальная) Риск 8
G/ Группы		Администратор: 1. Сохранять группы операций конфиденциальными (частота - очень редко, критичность максимальная) Риск 4
H/ Операции		Администратор: 1. Сохранять операции конфиденциальными (частота - очень редко, критичность максимальная) Риск 4
I/ Заказчики		Администратор: 1. Сохранять заказчиков конфиденциальными (частота - очень редко, критичность максимальная) Риск 4

После составления тепловой карты были выявлены риски: 24,39% незначительного, 17,07% среднего, 46,34% выше среднего и 12,20% высокого уровня. Выяснилось, что главный риск – недостоверность отображаемой в карточке двигателя информацией о ходе ремонта (например, завершенная операция была помечена как незавершенная), в частности отображение карточек двигателей с завершенным ремонтом в списке “В работе” и, наоборот, отображение незавершенных ремонтов в списке “Выполненные”. Также большую оценку рисков получила возможность назначения рабочих на заказ, так как в расписании не должно быть накладок, и возможность составления правильных отчетов о выполненных рабочими операциях. После были просуммированы оценки рисков по компонентам, что показано в табл. 4. Также были просуммированы значения рисков по атрибутам, что показано в табл. 5. И

просуммированы величины рисков по акторам, что показано в табл. 6. После оценки суммы величин рисков было решено составлять план тестирования по параметру акторы, так как это казалось самым пригодным параметром для составления по нему тест-плана, так как позволяло проводить тестирование по ролям, составлять пользовательские истории и т.д. Были выделены возможности, которые должны быть протестированы в начале и составлен тест-план. Критерием остановки тестирования было выбрано условие уменьшения суммы величин рисков для всей системы хотя бы на 15%. Проведено сценарное тестирование по ролям пользователей в системе [17-25]:

1. Тестировать возможности администратора
- 1.1 Тестирование назначение пользователю определенных ролей



- 1.2 Тестирование назначения операции в группу операций
- 1.3 Тестирование добавление/редактирования/удаления пользователей, групп, операций, заказчиков
- 1.4 Тестирование конфиденциальности данных при работе с отчетами, а также сохранение групп операций, операций, заказчиков в тайне
- 1.5 Тестирование невозможности назначения рабочего на заказ, если он уже занят
2. Тестировать возможности мастера
- 2.1 Тестирование возможности закончить работу, включая тестирование попадания ее в нужные списки при отображении
- 2.2 Тестирование возможности назначения рабочих в операцию
- 2.3 Тестирование возможности установить работе статус незавершенной (приостановленной)
3. Тестировать возможности начальника цеха
- 3.1 Тестирование возможности просматривать карточки выполненных/невыполненных работ, там должна отобра-

жаться корректная действительная информация о заказе

4. Тестировать возможности директора
- 4.1 Тестирование возможности просматривать сколько времени определенный рабочий потратил на определенные операции
- 4.2 Тестирование возможности просматривать какие операции выполняли конкретные рабочие в ходе выполнения заказа
- 4.3 Тестирование возможности скачивать отчет о заказах
5. Тестировать возможности, которые доступны всем пользователям системы
- 5.1 Тестирование возможности просматривать степень завершенности того или иного ремонта
- После проведения тестирования и устранения ошибок оценка вероятности рисков были пересмотрена, что видно по таблицам сравнения рисков до проведения АСС и после в табл. 2, табл. 3 и табл. 4.

Таблица 4. Таблица компонент и суммы рисков

Table 4. Table of components and amounts of risks

Компонент	Пояснение	Сумма	Сумма после АСС
Поиск	Поисковые строки на разных вкладках приложения	4	4
Карта ремонта	Карточка с информацией о ремонте	40	19
В работе	Список карт активных и приостановленных ремонтов	25	17
Выполненные	Список карт завершенных ремонтов	34	26
Отчеты	Вкладка для просмотра часов рабочих и информации по заказам	20	20
Пользователи	Вкладка администрирования данных пользователей	30	26
Группы	Вкладка администрирования групп работ	28	28
Операции	Вкладка администрирования возможных операций с двигателем	22	22
Заказчики	Вкладка администрирования данных о заказчиках работ	28	28

Таблица 5. Таблица атрибутов и суммы рисков

Table 5. Table of attributes and risk amounts

Атрибут	Пояснение	Сумма	Сумма после АСС
Простой	Интуитивно-понятные действия	124	99
Удобный	Минимизация операций для часто выполняемых действий	66	54
Доступный	Позволяет подключаться пользователям с разными ролями	14	14
Безопасный	Защищает информацию от различных угроз	27	23

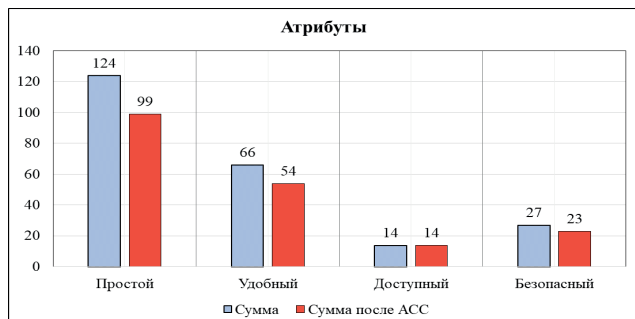
Таблица 6. Таблица акторов и суммы рисков по ним

Table 6. Table of actors and the amounts of risks for them

Роль	Сумма	Сумма после АСС
Админ	120	116
Мастер	42	21
Начальник цеха	32	16
Директор	22	22
Все	15	15

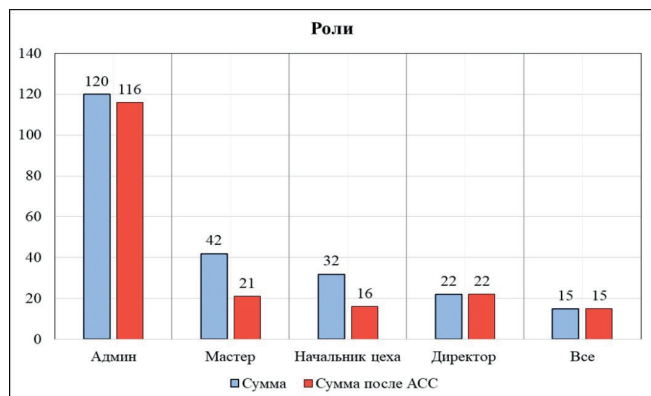


Всего сумма величин рисков стала меньше на 18%, наглядно изменение величин рисков до и после АСС анализа показано на рис. 1-3, где приведено сравнение рисков по отдельным атрибутам, акторам и компонентам.



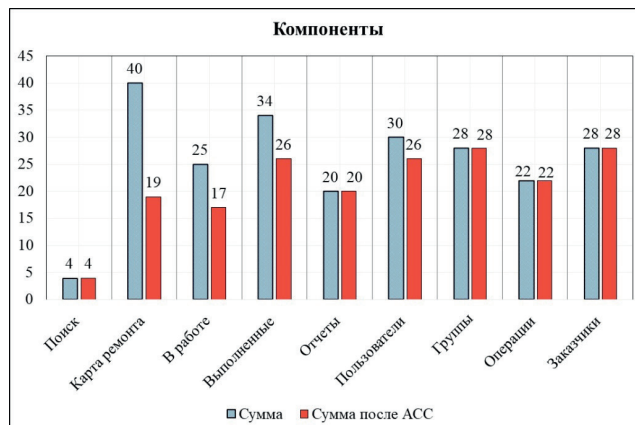
Р и с. 1. Сравнение суммы оценок рисков по атрибутам до и после АСС анализа

F i g. 1. Comparison of the sum of risk assessments by attributes before and after the ACC analysis



Р и с. 2. Сравнение суммы оценок рисков по ролям до и после АСС анализа

F i g. 2. Comparison of the sum of risk assessments by roles before and after the ACC analysis



Р и с. 3. Сравнение суммы оценок рисков по компонентам до и после АСС анализа

F i g. 3. Comparison of the sum of risk assessments by components before and after the ACC analysis

После получения результатов было принято решение прекратить тестирование, так как был выполнен критерий остановки тестирования. Всего было выявлено 13 ошибок при тестировании, что доказывает эффективность АСС анализа. В сравнении с другими методиками тестирования АСС выделяется тем, что затрагивает содержание информационной системы, а другие методики как RUP и IEEE нет, кроме того он позволяет приоритизировать тестирование, что в конечном итоге позволяет создавать эффективные тест-планы.

Заключение

Цель исследования состояла в том, чтобы составить эффективный план тестирования для приложения и найти и устранить самые значимые ошибки в информационной системе. Материалом исследования стала информационная система контроля учёта ремонта электродвигателей, методом исследования стала методика АСС анализа.

Оказалось, что классического применения АСС анализа может быть недостаточно и пришлось добавить дополнительное измерение в АСС анализ. Была составлена тепловая карта оценки рисков возможностей системы.

По тепловой карте был составлен план тестирования и проведены мероприятия по тестированию системы.

В ходе тестирования были выявлены и устранены ошибки, что позволило улучшить систему на 18%.



Список использованных источников

- [1] Forgács I., Kovács A. Software Testing Basics // Modern Software Testing Techniques. Berkeley, CA: Apress, 2024. P. 1-41. https://doi.org/10.1007/978-1-4842-9893-0_1
- [2] Сафиулин Р. З. Развитие технологий тестирования в образовании // Управление образованием: теория и практика. 2015. № 1(17). С. 139-149. EDN: TNGDAN
- [3] Карпунин А. А., Ганев Ю. М., Чернов М. М. Методы обеспечения качества при проектировании сложных программных систем // Надежность и качество сложных систем. 2015. № 2(10). С. 78-84. EDN: UIQABT
- [4] Галимова Е. Ю. Методика выбора автоматизированного, ручного и смешанного способа тестирования программного продукта, основанная на критериях качества // Известия Тульского государственного университета. Технические науки. 2019. № 7. С. 248-256. EDN: QBRJON
- [5] A Methodological Framework for Evaluating Software Testing Techniques and Tools / T. E. J. Vos [et al.] // 2012 12th International Conference on Quality Software. Xi'an, China: IEEE Press, 2012. Pp. 230-239. <https://doi.org/10.1109/QSIC.2012.16>
- [6] Полевщиков И. С., Чирков М. С., Леванов А. В. Автоматизированная система разработки тест-планов при проведении тестирования программного обеспечения // Инженерный вестник Дона. 2019. № 8(59). С. 29. EDN: HTKVBV
- [7] Синяя М. В., Ульянова Н. Д. Тестирование программного обеспечения: понятие, классификация, практика применения // Вестник цифровой экономики. 2020. № 1. С. 55-64. EDN: KYJZAW
- [8] Software Quality Assurance as a Service: Encompassing the quality assessment of software and services / S. Bernardo [et al.] // Future Generation Computer Systems. 2024. Vol. 156. P. 254-268. <https://doi.org/10.1016/j.future.2024.03.024>
- [9] Vavilov N. A. Computers As a Novel Mathematical Reality: III. Mersenne Numbers and Sums of Divisors // Doklady Mathematics. 2023. Vol. 107, issue 3. P. 173-204. <https://doi.org/10.1134/S1064562423700783>
- [10] Шакирова А. И., Хасьянов А. Ф., Даутов Э. Ф. Сокращение времени тестирования программного обеспечения // Современные наукоёмкие технологии. 2019. № 7. С. 104-109. EDN: XNHLRU
- [11] Вилданова К. И. Выбор метода тестирования программного обеспечения // Ученые записки УлГУ. Серия: Математика и информационные технологии. 2022. № 2. С. 31-37. EDN: ANTMVJ
- [12] Моисеев Д. А. Методология и процесс ручного тестирования // Надежность и качество сложных систем. 2017. № 3(19). С. 107-112. <https://doi.org/10.21685/2307-4205-2017-3-16>
- [13] Мартинов Г. М., Суханова Н. В., Козак Н. В. Повышение надежности программного обеспечения систем управления посредством автоматизации процесса тестирования // Приборы и системы. Управление, контроль, диагностика. 2009. № 11. С. 23-28. EDN: KZPVJJ
- [14] Лубков Н. В., Степанянц А. С., Викторова В. С. Надежные модели и анализ систем с защитой // Автоматика и телемеханика. 2018. № 7. С. 117-137. <https://doi.org/10.31857/S000523100000271-2>
- [15] Storm: program reduction for testing and debugging probabilistic programming systems / S. Dutta [et al.] // Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2019). New York, NY, USA : Association for Computing Machinery, 2019. P. 729-739. <https://doi.org/10.1145/3338906.3338972>
- [16] Zafar M. N., Afzal W., Enoiu E. Evaluating system-level test generation for industrial software: a comparison between manual, combinatorial and model-based testing // Proceedings of the 3rd ACM/IEEE International Conference on Automation of Software Test (AST '22). New York, NY, USA : Association for Computing Machinery, 2022. P. 148-159. <https://doi.org/10.1145/3524481.3527235>
- [17] Khaydarov Sh. A., Alikulov T. A. Mathematical Model of the Restored Technical System Reliability // International Journal of Computation and Applied Sciences. 2022. No. 5. P. 11-17. <https://doi.org/10.24411/2413-2101-2022-10501>
- [18] Khaneghah M. Z., Alzayed M., Chaoui H. Fault Detection and Diagnosis of the Electric Motor Drive and Battery System of Electric Vehicles // Machines. 2023. Vol. 11, issue 7. Article number: 713. <https://doi.org/10.3390/machines11070713>
- [19] Гагарина Л. Г., Букарев А. В. Метод автоматизированного тестирования устройств потребительской электроники с помощью удаленного вызова процедур и облачных сервисов // Известия высших учебных заведений. Электроника. 2023. Т. 28, № 5. С. 687-699. <https://doi.org/10.24151/1561-5405-2023-28-5-687-699>
- [20] Уткин А. В., Козлова Ю. Д. Облачные технологии в задаче автоматизации разработки и тестирования программного обеспечения // Международный журнал гуманитарных и естественных наук. 2024. № 4-5(91). С. 146-151. <https://doi.org/10.24412/2500-1000-2024-4-5-146-151>
- [21] Суркова Л. Е., Булычева С. А. Автоматизация процессов регрессионного тестирования при разработке и модернизации программного обеспечения // Информационно-технологический вестник. 2022. № 3(33). С. 73-81. EDN: IVDOKW
- [22] A Review of Automated Testing Approach for Software Regression Testing / Fransiskus Anindita Kristiawan Pramana Gentur Sutapa [et al.] // IOP Conference Series: Materials Science and Engineering. 2019. Vol. 846. Article number: 012042. <https://doi.org/10.1088/1757-899X/846/1/012042>
- [23] Sethi A. A review paper on levels, types & techniques in software testing // International Journal of Advanced Research in Computer Science. 2017. Vol. 8, issue 7. P. 269-271. <https://doi.org/10.26483/ijarcs.v8i7.4236>
- [24] Mayeda M., Andrews A. Chapter Two – Evaluating software testing techniques: A systematic mapping study // Advances in Computers. 2021. Vol. 123. P. 41-114. <https://doi.org/10.1016/bs.adcom.2021.01.002>



- [25] Isha S. S. Software Testing Techniques and Strategies // International Journal of Engineering Research and Applications. 2014. Vol. 4, issue 4. P. 99-102. URL: https://www.ijera.com/papers/Vol4_issue4/Version%209/S440999102.pdf (дата обращения: 15.11.2023).

Поступила 15.11.2023; одобрена после рецензирования 28.01.2024; принята к публикации 02.03.2024.

Об авторах:

Микишева Полина Алексеевна, студент Пермского филиала, ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики» (614070, Российская Федерация, г. Пермь, ул. Студенческая, д. 38), ORCID: <https://orcid.org/0009-0003-0246-5418>, mikisheva.p@yandex.ru

Мустафина Назгуль Ибрагимовна, студент Пермского филиала, ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики» (614070, Российская Федерация, г. Пермь, ул. Студенческая, д. 38), ORCID: <https://orcid.org/0000-0001-8617-659X>, nazgul-2003@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

- [1] Forgács I., Kovács A. Software Testing Basics. In: Modern Software Testing Techniques. Berkeley, CA: Apress; 2024. p. 1-41. https://doi.org/10.1007/978-1-4842-9893-0_1
- [2] Safiullin R.Z. Technology development of testing in education. *Education Management: Theory and Practice*. 2015;(1):139-149. (In Russ., abstract in Eng.) EDN: TNGDAN
- [3] Karpunin A.A., Ganey Yu.M., Chernov M.M. [Quality Assurance Methods for Designing Complex Software Systems]. *Reliability and Quality of Complex Systems*. 2015;(2):78-84. (In Russ., abstract in Eng.) EDN: UIQABT
- [4] Galimova E.Yu. Methods of selection an automated, manual and semi-automated method of testing a software product based on quality criteria. *News of the Tula State University. Technical sciences*. 2019;(7):248-256. (In Russ., abstract in Eng.) EDN: QBRJOH
- [5] Vos T.E.J., Marín B., Escalona M.J., Marchetto A. A Methodological Framework for Evaluating Software Testing Techniques and Tools. In: 2012 12th International Conference on Quality Software. Xi'an, China: IEEE Press; 2012. p. 230-239. <https://doi.org/10.1109/QSIC.2012.16>
- [6] Polevshchikov I.S., Chirkov M.S., Levanov A.V. Automated system for developing test plans for software testing. *Engineering journal of Don*. 2019;(8):29. (In Russ., abstract in Eng.) EDN: HTKVBV
- [7] Sinyaya M., Ulyanova N. Testing the software: concept, classification, practice of application. *Vestnik cifrovoy jekonomiki*. 2020;(1):55-64. (In Russ., abstract in Eng.) EDN: KYJZAW
- [8] Bernardo S., et al. Software Quality Assurance as a Service: Encompassing the quality assessment of software and services. *Future Generation Computer Systems*. 2024;156:254-268. <https://doi.org/10.1016/j.future.2024.03.024>
- [9] Vavilov N.A. Computers As a Novel Mathematical Reality: III. Mersenne Numbers and Sums of Divisors. *Doklady Mathematics*. 2023;107(3):173-204. <https://doi.org/10.1134/S1064562423700783>
- [10] Shakirova A.I., Khasyanov A.F., Dautov E.F. Reduction of testing time software. *Modern high technologies*. 2019;(7):104-109. (In Russ., abstract in Eng.) EDN: XNHLRU
- [11] Vildanova K.I. Choosing a software testing method. *Uchenyye zapiski UIGU. Seriya "Matematika i informatsionnyye tekhnologii"*. 2022;(2):31-37. (In Russ., abstract in Eng.) EDN: ANTMVJ
- [12] Moiseev D.A. [Manual Testing Methodology and Process]. *Reliability and Quality of Complex Systems*. 2017;(3):107-112. (In Russ., abstract in Eng.) <https://doi.org/10.21685/2307-4205-2017-3-16>
- [13] Martinov G.M., Sukhanova N.V., Kozak N.V. Increasing reliability of the control systems software with help of automating the process of testing. *Instruments and Systems: Monitoring, Control, and Diagnostics*. 2009;(11):23-28. (In Russ., abstract in Eng.) EDN: KZPVJJ
- [14] Lubkov N., Stepanyants A., Viktorova V. Reliability models and analysis of systems with protection. *Avtomatika i telemekhanika = Automation and Remote Control*. 2018;(7):117-137. (In Russ., abstract in Eng.) <https://doi.org/10.31857/S000523100000271-2>
- [15] Dutta S., Zhang W., Huang Z., Misailovic S. Storm: program reduction for testing and debugging probabilistic programming systems. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2019). New York, NY, USA: Association for Computing Machinery; 2019. p. 729-739. <https://doi.org/10.1145/3338906.3338972>
- [16] Zafar M.N., Afzal W., Enoiu E. Evaluating system-level test generation for industrial software: a comparison between manual, combinatorial and model-based testing. In: Proceedings of the 3rd ACM/IEEE International Conference on Automation of Software Test (AST'22). New York, NY, USA: Association for Computing Machinery; 2022. p. 148-159. <https://doi.org/10.1145/3524481.3527235>
- [17] Khaydarov Sh.A., Alikulov T.A. Mathematical Model of the Restored Technical System Reliability. *International Journal of Computation and Applied Sciences*. 2022;(5):11-17. <https://doi.org/10.24411/2413-2101-2022-10501>
- [18] Khaneghah M.Z., Alzayed M., Chaoui H. Fault Detection and Diagnosis of the Electric Motor Drive and Battery System of Electric Vehicles. *Machines*. 2023;11(7):713. <https://doi.org/10.3390/machines11070713>



- [19] Gagarina L.G., Bukarev A.V. A method for automated testing of consumer electronics devices using remote procedure calls and cloud services. *Proceedings of Universities. Electronics*. 2023;28(5):687-699. (In Russ., abstract in Eng.) <https://doi.org/10.24151/1561-5405-2023-28-5-687-699>
- [20] Utkin A.V., Kozlova Yu.D. Cloud technologies in the task of automating development and testing software. *International Journal of Humanities and Natural Sciences*. 2024;(4-5):146-151. (In Russ., abstract in Eng.) <https://doi.org/10.24412/2500-1000-2024-4-5-146-151>
- [21] Surkova L.E., Bulycheva S.A. Automation of regression testing processes in software development and modernization. *Information technology Vestnik*. 2022;(3):73-81. (In Russ., abstract in Eng.) EDN: IVDOKW
- [22] Fransiskus Anindita Kristiawan Pramana Gentur Sutapa, et al. A Review of Automated Testing Approach for Software Regression Testing. *IOP Conference Series: Materials Science and Engineering*. 2019;846:012042. <https://doi.org/10.1088/1757-899X/846/1/012042>
- [23] Sethi A. A review paper on levels, types & techniques in software testing. *International Journal of Advanced Research in Computer Science*. 2017;8(7):269-271. <https://doi.org/10.26483/ijarcs.v8i7.4236>
- [24] Mayeda M., Andrews A. Chapter Two – Evaluating software testing techniques: A systematic mapping study. *Advances in Computers*. 2021;123:41-114. <https://doi.org/10.1016/bs.adcom.2021.01.002>
- [25] Isha S.S. Software Testing Techniques and Strategies. *International Journal of Engineering Research and Applications*. 2014;4(4):99-102. Available at: https://www.ijera.com/papers/Vol4_issue4/Version%209/S440999102.pdf (accessed 15.11.2023).

Submitted 15.11.2023; approved after reviewing 28.01.2024; accepted for publication 02.03.2024.

About the authors:

Polina A. Mikisheva, student of the Perm Branch of the National Research University Higher School of Economics (38 Studencheskaya St., Perm 614070, Russian Federation), ORCID: <https://orcid.org/0009-0003-0246-5418>, mikisheva.p@yandex.ru

Nazgul I. Mustafina, student of the Perm Branch of the National Research University Higher School of Economics (38 Studencheskaya St., Perm 614070, Russian Federation), ORCID: <https://orcid.org/0000-0001-8617-659X>, nazgul-2003@mail.ru

All authors have read and approved the final manuscript.

