

Нарышкин Константин Викторович

**МЕТОД ОЦЕНКИ КАЧЕСТВА КОМПЬЮТЕРНЫХ ЭЛЕМЕНТОВ
СИСТЕМЫ УПРАВЛЕНИЯ ПРИ ПЕРЕНОСЕ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ НА АЛЬТЕРНАТИВНЫЕ АППАРАТНЫЕ
ПЛАТФОРМЫ**

2.3.1. Системный анализ, управление и обработка информации, статистика

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург – 2025

Работа выполнена в Автономной некоммерческой организации дополнительного профессионального образования «Научно – образовательный центр ВКО «Алмаз – Антей» им. академика В.П. Ефремова»

Научный руководитель: доктор технических наук,
Новиков Владимир Александрович

Официальные оппоненты: **Билятдинов Камиль Закирович**,
доктор технических наук, доцент
Акционерное общество «Научно-исследовательский
институт «Масштаб», г. Санкт-Петербург, советник
генерального директора

Болбаков Роман Геннадьевич,
кандидат технических наук, доцент,
МИРЭА – Российский технологический университет,
кафедра инструментального и прикладного
программного обеспечения Института информационных
технологий, кафедра инструментального и прикладного
обеспечения, заведующий кафедрой

Ведущая организация: Федеральное государственное бюджетное учреждение
науки «Санкт-Петербургский Федеральный
исследовательский центр Российской академии наук»,
г. Санкт-Петербург

Защита состоится 10 сентября 2025 года в 14.00 на заседании объединенного диссертационного совета 99.2.038.03, созданного на базе Федерального государственного бюджетного образовательного учреждения высшего образования «Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова», Федерального государственного автономного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет аэрокосмического приборостроения», Федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» по адресу: Санкт-Петербург, пр. Большевиков, д. 22, корп. 1, ауд. 554/1.

С диссертацией можно ознакомиться в библиотеке СПбГУТ по адресу Санкт-Петербург, пр. Большевиков, д. 22, корп. 1 и на сайте www.sut.ru.

Автореферат разослан 3 июля 2025 года.

Ученый секретарь
диссертационного совета 99.2.038.03,
канд. техн. наук, доцент

А.Г. Владыко

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. Устойчивое функционирование предприятий (в том числе, предприятий машиностроения) связана с непрерывным обновлением действующих сложных технических систем. Совершенствование сложных систем управления достигается постоянной разработкой и внедрением новых программно-аппаратных комплексов (ПАК), которые не могут быть разработаны и внедрены единовременно из-за высокой сложности. Создание технологического суверенитета предполагает такой процесс модернизации (в частности импортозамещение), который позволит за минимальное время заменить действующие ПАК в системе управления (компьютерные элементы системы управления) без значительной потери основных характеристик качества.

Основой процесса модернизации является создание нового или использование существующего контролируемого аппаратного обеспечения в компьютерных элементах системы управления (КЭ СУ). Использование существующего аппаратного обеспечения требует меньше ресурсов, но ограничено тем перечнем компонентов, которые доступны для приобретения. Как при создании нового программного обеспечения, так и при переносе существующего на альтернативные аппаратные платформы (ААП) возникает потребность в многокритериальной оценке качества КЭ СУ и разработке на этой основе рекомендаций по управлению процессом модернизации, получаемой сложной системы управления.

Трудности в процессе оценивания при модернизации систем управления за счет использования ААП связаны с необходимостью учитывать неизвестные эмерджентные свойства, которые обусловлены аппаратными и программными факторами. Точность оценки достигается тем, что расчеты показателей качества применяются на всех возможных этапах разработки и внедрения КЭ СУ. При модернизации систем управления за счет переноса программного обеспечения на ААП отсутствует большой объем статистических данных и имеется большое количество неизвестных аппаратно-программных факторов.

Отсюда, возникает противоречие, которое заключается в том, что с одной стороны в условиях санкций и импортозамещения требуется модернизировать действующие системы за счет переноса программного обеспечения на ААП, а с другой существующие алгоритмы поиска ошибок, модели и методы оценки качества вычислительных систем не обеспечивают требуемый уровень точности для принятия решений о внедрении получаемых программно-аппаратных комплексов в действующие системы управления.

Степень разработанности темы исследования. Рассмотрены работы по моделям качества Боэма, Липаева В.В., Щенникова А.Н., Катылюк И.С, Тацiana Н. Кудо, по моделям надежности Балыбердина В.А., Василенко Н.В., Муравьева К.А., по моделям безопасности Ибрагимова Б.Г.

Проведен анализ работ, описывающие: измерительные методы программных (Звездин С.В., Лукин В.Н., Каушан В.В., Ермаков М.К.) и аппаратных компонент (Э. Хамадулин) информационной системы, регистрационные методы, в частности, методы тестирования (Коташев А.А., Чупилко М.Л., Золотухина Е.Б., Владимиров М.А., Данилов А.Д.), методы анализа дерева отказов (Берман А.Ф.), расчетные методы (Тацiana Н. Кудо, Балыбердин В.А., Неборский С.Н., Ковтун Н.И., Билятдинов К.З.)

Проведен анализ применения метода анализа исходного и исполняемого кода в процессах: разработки (Лаврищева Е.М., Иванников В.П., Белеванцев А. А.), обеспечения информационной безопасности (Новиков В.А., Буйневич М. В., Израилов К.А., Диасамидзе С.В.), проведения судебной экспертизы (Тарасов Д.А.)

Цель работы заключается в повышении точности оценки качества сложных систем управления в процессе модернизации, которая обеспечивается переносом существующего программного обеспечения на альтернативные аппаратные платформы за счет совершенствования моделей и методов оценки качества компьютерных элементов.

Для достижения поставленной цели в работе решены следующие **задачи**:

1. Провести анализ моделей качества и методов оценки качества вычислительных систем в составе систем управления.
2. Разработать модель качества КЭ СУ, показатели которой зависят от параметров исполняемого кода.
3. Разработать алгоритм поиска аппаратно-зависимых ошибок в исполняемом коде, используя сравнительный анализ формальных моделей программных компонент.
4. Разработать методику оценки влияния найденных ошибок в исполняемом коде на характеристики качества, а также комплект программ по реализации разработанной методике.
5. Разработать метод оценки характеристик качества на основе информации об ошибках, полученной в результате анализа исполняемого кода КЭ СУ.
6. Провести экспериментальную проверку полученных результатов исследования. Выработать рекомендации и предложения по использованию анализа исполняемого кода в процессе принятия решений о возможной модернизации КЭ СУ.

Научная задача, решаемая в работе, состоит в разработке научно-методического аппарата, повышающего точность оценки качества компьютерных элементов сложной системы управления в процессе модернизации аппаратного и программного обеспечения.

Научная новизна результатов работы:

– усовершенствована аналитическая модель качества ВС, которая в отличие от существующих учитывает аппаратно-независимые ошибки, их качество и количество в исполняемом коде;

– разработан алгоритм поиска ошибок в исполняемом коде, который основан на существующих методах обратного проектирования, что позволяет обнаружить ошибки в готовом образце системы без технической документации и исходных кодов ПО;

– разработана методика оценки влияния ошибки в ПО, которая в отличие от существующих, использует аппарат нечеткой логики, что позволяет решить проблему отсутствия статистических данных при переносе ПО на ААП;

– разработан метод оценки качества КЭ СУ, который в отличие от существующих, использует анализ исполняемого кода для того, чтобы осуществить поиск ошибок, принять решение о возможной модификации ПО и установить контроль над функциональными компонентами системы.

Теоретическая значимость работы заключается в разработке моделей и методов, являющихся дополнением в методологии оценки базовых характеристик качества (функциональная пригодность, надежность, производительность) КЭ СУ с учетом полного или частичного отсутствия исходных кодов.

Практическая значимость работы состоит в том, что её результаты в условиях отсутствия технической информации о КЭ СУ (техническая документация, исходные коды и др.) позволяют рассчитать многокритериальную оценку характеристик качества на основе готового образца системы. Такой подход позволяет проводить оценивание качества систем, которые получены в результате переноса ПО на ААП.

Практическая значимость и новизна подтверждаются тем, что на основе предложенного метода оценки влияния ошибки на систему разработан комплект программ, защищенных авторским свидетельством.

Внедрение результатов диссертационной работы. Полученные результаты использованы в учебном процессе Института кибербезопасности и цифровых технологий на кафедре «Интеллектуальные системы информационной безопасности» МИРЭА – Российский технологический университет (РТУ МИРЭА), а также в научной и практической деятельности акционерного общества «АСТ» (АО «АСТ»).

Методы исследования. Решение поставленных в работе задач выполнено с использованием методов модельно-ориентированного системного инжиниринга, теории сетей Петри, теории массового обслуживания, теории вероятностей, теории нечеткой логики, теории логико-вероятностного исчисления. Для исследования эффективности разработанных методов и алгоритмов проводилась экспериментальная проверка на встроенном программном обеспечении сетевых устройств и имитационное моделирование на ЭВМ.

Основные положения диссертации, выносимые на защиту

1. Аналитическая модель качества компьютерных элементов системы управления, представляющая математическую зависимость оценки качества от параметров исполняемого кода;
2. Методика оценки влияния ошибки в программном обеспечении на характеристики качества компьютерных элементов системы управления, использующая аппарат нечеткой логики для учета разнородных метрических характеристик воспроизведения ошибки;
3. Метод оценки качества компьютерных элементов системы управления, основанный на анализе моделей исполняемого кода и алгоритме поиска программных ошибок.

Достоверность полученных результатов подтверждена результатами моделирования и аналитическими расчётами. Результаты, полученные в ходе выполнения исследования, не противоречат ранее полученным данным, опубликованным в открытых источниках.

Апробация работы. Основные положения диссертации докладывались и обсуждались на: Научно-практической конференции «Цифровые технологии и их приложения» (г. Уфа, 2021); V международная научная конференция «Приоритетные направления инновационной деятельности в промышленности». (г. Казань, 2021); Всероссийская научно-практическая конференция «Актуальные тренды цифровой трансформации промышленных предприятий» (г. Казань, 2022); Международная научная конференция «Актуальные проблемы прикладной информатики в образовании, экономике, государственном и муниципальном управлении» (г. Барнаул, 2022); XLVIII международная научно-практическая конференция. «Российская наука в современном мире» (г. Москва, 2022); VII, VIII научно-техническая конференция «Математическое моделирование, инженерные расчеты и программное обеспечение для решения задач ВКО» (г. Москва, 2022-2023); I и III национальная научно-

практическая конференция «Кибербезопасность: технические и правовые аспекты защиты информации» (г. Москва, 2023-2024).

Публикации. Результаты исследований по теме диссертации опубликованы в 15 научных работах, из которых – 2 статьи в соавторстве, 1 статья в научно-практическом журнале, 4 статьи – в рецензируемых изданиях, рекомендованных ВАК Минобрнауки России, 9 – в сборниках трудов научных конференций. Получено свидетельство о государственной регистрации программы для ЭВМ №2023669264.

Соответствие диссертации паспорту научной специальности. Диссертационная работа соответствует пунктам 3 и 11 паспорта научной специальности 2.3.1. Системный анализ, управление и обработка информации, статистика. Проведенные исследования соответствуют формуле специальности.

Личное участие автора состоит в том, что все научные результаты получены автором самостоятельно. В работе предлагается использовать анализ исполняемого кода для определения параметров программных компонент компьютерных элементов систем управления, разработана и реализована методика оценки влияния ошибок в программном коде на систему в условиях отсутствия статистических данных и предложен метод оценки качества компьютерных элементов систем управления по базовым характеристикам качества.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка использованных источников (111 наименований). Общий объем работы 158 страниц, в том числе 143 страницы основного текста, включая 58 рисунков и 17 таблиц.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность темы диссертации, сформулированы цель, задачи, новизна исследования, представлена степень достоверности, практическая значимость полученных результатов и положения, выносимые на защиту.

В **первой главе** диссертации выполнен аналитический обзор существующих моделей и методов оценки качества вычислительных систем, которые основаны на разнородных характеристиках. В отечественной и зарубежной научной литературе исследуются модели Бозма, МакКола, FURPS, Дроми, Гилба, GQM. Иерархическая модель качества, представленная в работах Липаева В.В., является основой в области оценки качества сложных программных систем, а аналитическая модель, разработанная Лаврищевой Е.М. позволяет проводить расчеты отдельных показателей

качества (ПК). Недостатком этих моделей является отсутствие механизмов прикладного применения. Также исследования имеют большую асимметрию в сторону надежности и безопасности, хотя функциональная пригодность и производительность не менее важны.

Развитие моделей оценки качества ПАК направлено на сокращение метрик, определяемых экспертными методами. Так как основным источником данных для оценки внутреннего качества является сама система и техническая информация о ней, то её параметризация позволяет построить внутреннюю модель качества с использованием расчетных методов. Такая модель в большей степени объективна и уточняется не за счет экспертного знания, а за счет измеряемых величин. В тоже время не существуют комплексной формальной модели расчета разрозненных характеристик качества, что является проблемой при формировании единой оценки качества. Отсутствие многокритериальной оценки не позволяет управлять процессом внедрения компьютерных элементов в сложную систему.

Сформулированная проблема связана с тем, что существующие методы оценки качества не обладают достаточной точностью в расчётах отдельных ПК для тех систем, где оказывают влияние такие факторы как:

- неделимость системы;
- полное или частичное отсутствие технической документации;
- применение систем защиты информации;
- отсутствие поддержки производителя;
- отсутствие исходных кодов на программные компоненты;
- применение нестандартизированных технологий.

Так, например, измерительные методы оценки качества ПАК основаны на метрическом анализе программного кода. К недостаткам таких методов относится низкая связность большого числа количественных показателей, которые в одних случаях могут указывать на проблемы с качеством, а в других не влияют на него. Также при отсутствии исходных кодов методы измерения не адаптированы для исполняемого или промежуточного кодов.

Методы оценки качества ПАК прямо или косвенно направлены на обнаружение и анализ преднамеренных или случайных ошибок, оставленных в коде системы. Соответственно точность оценки качества зависит от количества найденных ошибок в системе и степени их влияния на базовые характеристики качества. Так как **объектом исследования** является оценивание качества КЭ сложной СУ в процессе модернизации программного (ПО) и аппаратного обеспечения (АО), то проведен анализ методов

оценки в случаях полного цикла разработки (исходное состояние системы) и в случаях, когда исходные коды отсутствуют (некоторые состояния системы при переносе ПО на альтернативные аппаратные платформы (ААП)). Установлено, что при переносе ПО на ААП средняя вероятность обнаружения всех ошибок снижается на 27%, так как зависит от вероятности обнаружения аппаратно-зависимых ошибок. Расчёты показывают, что систематическая погрешность методов оценки качества при исходном состоянии системы составляет 10 %, а в случаях переноса ПО на ААП погрешность возрастает до 28 %.

Поэтому, требуется усовершенствовать аналитическую модель качества вычислительной системы, которая учитывает ошибки в исполняемом коде; разработать алгоритм анализа исполняемого кода при переносе ПО на ААП, который позволит повысить вероятность обнаружения ошибок; разработать метод оценки влияния ошибки на характеристики качества; усовершенствовать существующие методы оценки качества КЭ СУ за счет разработанной аналитической модели качества, а также алгоритма поиска ошибок и метода оценки влияния. Разработанные модель, алгоритм и методы повышают точность оценки качества, за счет снижения систематической погрешности, до значений, полученных на этапах разработки исходной системы.

Как следствие поставленных задач исследований по разработке аналитической модели качества и метода оценки качества КЭ СУ, **во второй главе** представлена математическая модель качества КЭ СУ, которая основана на метриках исполняемого кода программных компонент. Целесообразно разделить систему на аппаратные и программные компоненты, а затем создать класс моделей под каждую характеристику качества. Модельно-ориентированный подход применяется в системном инжиниринге для описания и представления технических систем. Сбалансированный комплекс частных моделей образует метамоделли, которые описывают сложные системы.

Параметризация моделей основана на определении метрик a_k сущностей E_i , которые представляются в виде количественных величин. Разработана иерархическая модель качества, в которой выделено 3 характеристики качества и 12 ПК.

Функциональная пригодность (q_1) определяется функциональной полнотой (1), корректностью (3), точностью (5), целесообразностью (7), согласованностью (9):

Функциональная полнота

$$a_1^f = \frac{\sum_{i=1}^{|F^c|} \widehat{f}_i^c}{|F^m|}, \quad (1)$$

где F^c – это множество реализованных функций каждым элементом которого являются реализованные программные функции $\{f_1^c, \dots, f_n^c\}$, F^m – множество функций,

которые могут быть реализованы без ошибок на множестве программных функций $\{f_1^m, \dots, f_n^m\}$.

$$\widehat{f}_i^c = \text{erfc} \left(\sum_{k=1}^n w_k^f \right), \quad (2)$$

где w_k^f – оценка влияния программной ошибки на функциональную пригодность
Корректность

$$a_2^f = 1 - \frac{|F^m \setminus F^c|}{|F^m|}, \quad (3)$$

Точность

$$a_3^f = \frac{\sum_{i=1}^{|F^m|} (I_i^c(D) - I_i^m(D))}{|F^m|}, \quad (4)$$

где D – это множество $\{d_1, \dots, d_i\}$ входных значений в программную функцию.

Пусть $Dr \in D$, такое что индикаторная функция $I_i^m(d_i) = 1$, тогда

$$I_i^c(d_i) = \begin{cases} 1, & \text{если } d_i \in Dr \\ 0, & \text{если } d_i \notin Dr \end{cases} \quad (5)$$

Целесообразность

$$a_4^f = \frac{\sum_{i=1}^{|F^a|} f_i^a}{|F^m|}, \quad (6)$$

где F^a – это множество значений избыточных $\{f_1^a, \dots, f_n^a\}$ реализованных программных функций.

$$f_i^a = \begin{cases} \widehat{f}_i^c, & f_i^c \in F^m \\ 0, & f_i^c \notin F^m \end{cases} \quad (7)$$

Согласованность

$$a_5^f = \frac{\sum_{i=1}^{|F^c|} \widehat{f}_i^s}{|F^c|}, \quad (8)$$

где $\widehat{f}_i^s$ – оценка стандартности i -й программной функции.

$$\widehat{f}_i^s = \begin{cases} \widehat{f}_i^c, & f_i^c \in F^s \\ 0, & f_i^c \notin F^s \end{cases}, \quad (9)$$

где F^s – множество стандартных программных реализаций.

Так как функциональная пригодность основана на однородных метриках, то общая оценка является сверткой всех показателей качества a_k^f . В силу того, что значимость показателей разная, то необходимо ввести весовые коэффициенты ω_k^f . Тогда расчет характеристики q_1 записывается следующей формулой:

$$\begin{aligned}
q_1 &= \sum_{k=1}^5 \omega_k^f * a_k^f = \\
&= \omega_1^f * \frac{\sum_{i=1}^{|F^c|} \widehat{f}_i^c}{|F^m|} + \omega_2^f * \left(1 - \frac{|F^m \setminus F^c|}{|F^m|}\right) + \omega_3^f * \frac{\sum_{i=1}^{|F^m|} (I_i^c(D) - I_i^m(D))}{|F^m|} + \\
&\quad + \omega_4^f * \frac{\sum_{i=1}^{|F^a|} f_i^a}{|F^m|} + \omega_5^f * \frac{\sum_{i=1}^{|F^c|} \widehat{f}_i^s}{|F^c|}
\end{aligned} \tag{10}$$

Надежность КЭ СУ определяется безотказностью (12), устойчивостью к ошибкам (13), восстанавливаемостью (14). Пусть система обладает множеством отказов $M = \{m_1, \dots, m_i\}$ с m^* уникальных типов отказов, которые вызваны ошибками в ПО. Тогда для расчетов ПК надежности имеет место такая характеристика отказа в системе, которая зависит от программных ошибок (11).

$$\widehat{m}_i = \begin{cases} \sum_{k=1}^n w_k^d, & \sum_{k=1}^n w_k^d < 1, \\ 1, & \sum_{k=1}^n w_k^d > 1 \end{cases} \tag{11}$$

где w_k^d - оценка влияния программной ошибки на надежность системы.

Безотказность

$$a_1^d = \left(\sum_{i=1}^n \left(\frac{T_i - \min(T)}{\max(T) - \min(T)} \right) \right) / n, \tag{12}$$

где $T = \{T_1, \dots, T_i\}$ - множество наработок до отказа.

$$T_i = \frac{t_i^M + t_i^I}{\widehat{m}_i} \text{ при } \widehat{m}_i \neq 0,$$

где t_i^M – это время до инициализации ошибочного программного кода, а t_i^I - время инициализации и выполнения ошибочного программного кода

Устойчивость к ошибкам

$$a_2^d = \frac{m^*}{\sum_{i=1}^n \widehat{m}_i} \text{ при } \widehat{m}_i \neq 0, \tag{13}$$

где m^* - количество типов отказов.

Восстанавливаемость

$$a_3^d = \left(\sum_{i=1}^n \left(1 - \frac{TR_i - \min(TR)}{\max(TR) - \min(TR)} \right) \right) / n, \tag{14}$$

где $TR = \{TR_1, \dots, TR_i\}$ - значение восстанавливаемости после i -го отказа.

$$TR_i = \frac{t_i^R + t_i^M}{\widehat{m}_i} \text{ при } \widehat{m}_i \neq 0,$$

где t_i^R – это время, за которое выполняется программный код при восстановлении, а t_i^M – время, которое требуется для вмешательства из вне.

Свертка (15) характеристики качества – надежность (q_2) осуществима только при условии, что введены весовые коэффициенты ПК ω_k^d .

$$\begin{aligned}
 q_2 &= \sum_{k=1}^3 \omega_k^d * a_k^d = \\
 &= \omega_1^d * \left(\sum_{i=1}^n \left(\frac{T_i - \min(T)}{\max(T) - \min(T)} \right) \right) / n + \omega_2^d * \frac{m^*}{\sum_{i=1}^n m_i} + \\
 &\quad + \omega_3^d * \left(\sum_{i=1}^n \left(1 - \frac{TR_i - \min(TR)}{\max(TR) - \min(TR)} \right) \right) / n
 \end{aligned} \tag{15}$$

Метрики модели ПАК в виде системы массового обслуживания (СМО) являются ЭПК, что позволяет рассчитать ПК производительности: пропускная способность (18), время отклика (21), ресурсоемкость по памяти (22), ресурсоемкость по процессору (23).

Расчет (18) и (21) основан на моделях СМО с отказами. СМО с ограниченной очередью моделируют компоненты КЭ СУ, которые обладают буфером данных. Буферы данных ограничены аппаратными возможностями системы, поэтому СМО с неограниченной очередью не подходят для моделирования. На рисунке 1 представлена функция обработки сообщений (данных), которая обладает ограниченным буфером.

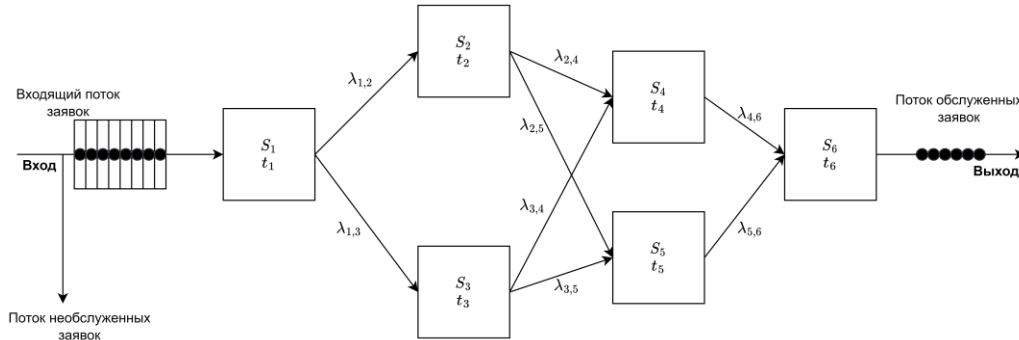


Рисунок 1 – Граф состояний функции обработки сообщений с ограниченным буфером

Одноканальная СМО с ограниченной очередью работает по схеме гибели и размножения с конечным числом состояний S_i . В результате исследований получены формулы, которые описывают расчеты пропускной способности и время отклика по отдельным программным компонентам системы.

Предельная вероятность при состоянии S_0 :

$$p_{0i} = \frac{1 - \rho_i}{1 - \rho_i^{m+2}} \tag{16}$$

$$\rho_i = \frac{\lambda}{\hat{\mu}}, \tag{17}$$

где λ - интенсивность перехода из S_k в S_{k+1} , а $\hat{\mu} = f(w_k^p)$ – это функция интенсивности обработки сообщений, которая зависит от оценки влияния ошибки на производительность системы w_k^p .

Относительная пропускная способность

$$a_1^p = \sum_{i=1}^n \left(\frac{1 - \rho_i^{m+1}}{1 - \rho_i^{m+2}} \right) \quad (18)$$

Среднее число заявок, находящихся под обслуживанием:

$$Ls_{ij} = 1 - p0_i \quad (19)$$

Среднее число заявок, стоящих в очереди:

$$Lq_i = \rho_i^2 \frac{1 - \rho_i^m (m + 1 - m\rho_i)}{(1 - \rho_i)(1 - \rho_i^{m+2})} \quad (20)$$

Время отклика

$$a_2^p = \sum_{i=1}^n \frac{Ls_i + Lq_i}{\lambda} \quad (21)$$

Модели программной функции в виде одноканальной СМО без отказов, позволяют рассчитать наиболее вероятные состояния функции, которые требуют анализа с точки зрения потребления ресурсов. После определения наиболее вероятных состояний выделяются те программные участки кода, которые отвечают за выделения памяти (на стеке и кучи), проводится подсчет затрачиваемой памяти (V^m), а также определяется количество процессорного времени на обработку вероятного состояния (C^l).

Ресурсоемкость по памяти

$$a_3^p = \frac{\sum_{i=1}^n V_i^m}{V}, \quad (22)$$

где V_i^m – объем затраченной памяти в i -ом программном компоненте, V – объем всей доступной памяти.

Ресурсоемкость по процессору

$$a_4^p = \frac{\sum_{i=1}^n C_i^l}{C}, \quad (23)$$

где C_i^l – объем затраченного процессорного ресурса в i -ом программном компоненте, C – объем всего ресурса процессора.

Как и для других характеристик качества для производительности разработана формула расчета по всем представленным ПК (25) с учетом весовых коэффициентов ω_k^p .

$$q_3 = \sum_{k=1}^4 \omega_k^p * a_k^p =$$

$$= \omega_1^p * \sum_{i=1}^n \left(\frac{1 - \rho_i^{m+1}}{1 - \rho_i^{m+2}} \right) + \omega_2^p * \frac{\sum_{i=1}^n Ls_i + Lq_i}{\sum_{i=1}^n Ls'_i + Lq'_i} + \omega_3^p * \left(1 - \frac{\sum_{i=1}^n V_i^m}{V} \right) +$$

$$+ \omega_4^p * \left(1 - \frac{\sum_{i=1}^n C_i^l}{C} \right) \quad (24)$$

Таким образом, получена аналитическая модель качества вычислительной системы, которая позволяет рассчитать базовые характеристики (q_1, q_2, q_3) КЭ СУ.

В третьей главе разрабатывался метод оценки показателей качества КЭ СУ. Метод включает в себя: алгоритм поиска ошибок в программных компонентах, метод оценки влияния найденных ошибок на характеристики качества; аналитическую модель качества, которая позволяет рассчитать многокритериальную оценку.

Алгоритм поиска ошибок основан на априорных моделях программных компонент системы и их сравнении с моделями, построенными на основе анализа исполняемого кода. На рисунке 2 представлен алгоритм поиска ошибок в программных компонентах готового образца системы.

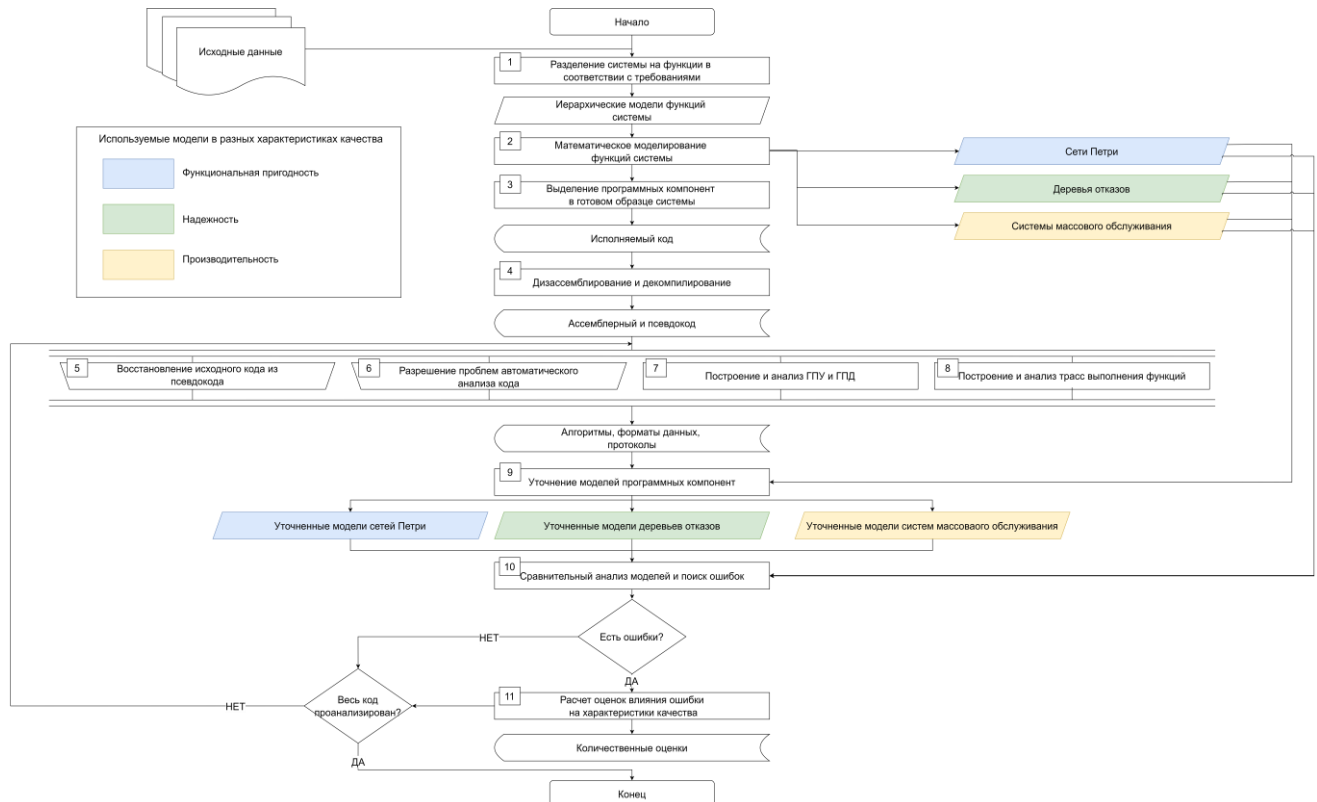


Рисунок 2 – Блок-схема алгоритма поиска ошибок

Согласно алгоритму, система делится в соответствии с функциями (*FBS*), которые она выполняет, и компонентами (*PBS*), из которых она состоит (этап 1). Так как моделирование (этап 2) ограничено отсутствием информации о системе (факторы внешней среды), то полнота моделей *FBS* и *PBS* ограничена доступной информацией. Поэтому, в процессе поиска ошибок используются методы восстановления информации о системе (анализ исполняемого кода и восстановление информации об аппаратных компонентах) (этап 3-9). Таким образом, применение методов восстановления информации о функциях и компонентах системы обеспечивает модели *FBS* и *PBS* достаточной полнотой для поиска ошибок путем сравнительного анализа априорных и уточненных моделей (этап 10).

Используя парадигму вычислений на сетях Петри, открывается возможность к моделированию существующих систем через цепочку трансляций от бинарного кода через язык *assembler* на язык сетей Петри.

Детализация модели требует изучения системы или знания о работе программных компонент. Детализировать модель программных функций возможно до инструкций ассемблера.

Сравнительный анализ свойств моделей реализованных и требуемых функций, позволяет выявлять ошибки.

Согласно алгоритму необходимо построить априорную модель функции, а затем выделить программную функцию из исполняемого кода системы и уточнить априорную модель. С помощью инструментов дезассемблирования проведен анализ функции и построена уточненная сеть Петри (рис.3).

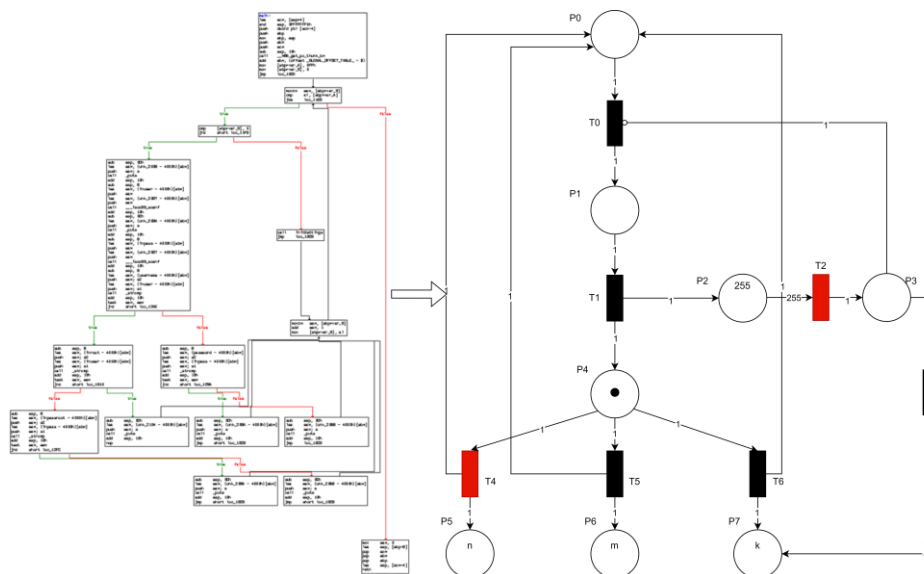


Рисунок 3 – Уточнение модели функции «аутентификации и авторизации»

Сравнительный анализ моделей экспериментальной функции позволил выявить ошибку целочисленного переполнения счетчика, от которого зависит условие перехода при первичной авторизации в системе.

Разбиение системы на требования, функции и компоненты позволяет формально описывать отказы в виде графическо-логических построений. Анализ деревьев отказов, построенных на основе программных компонент и с учетом аппаратной платформы, способен выявлять вероятные цепочки событий, которые приводят к отказам.

Пусть A – множество базовых событий A_i вероятность наступления которых $P(A_i)$, тогда множество промежуточных событий B_i рассчитывается как произведение вероятностей базовых событий входящих в логический оператор «И».

$$P(A_1 A_2 \dots A_n) = \prod_{i=1}^n P(A_i) \quad (25)$$

Вход базовых событий в логический оператор «ИЛИ» определяет вероятность промежуточного события по формуле суммы конечного числа совместных событий.

$$P\left(\sum_{i=1}^n A_i\right) = \sum_i P(A_i) - \sum_{i,j} P(A_i A_j) + \sum_{i,j,k} P(A_i A_j A_k) + \dots + (-1)^{n-1} P(A_1 A_2 \dots A_n), \quad (26)$$

где суммы распространяются на различные значения индексов i, j, k , и их комбинаций.

Действуя по аналогии, рассчитываются вероятности отказов компонент $P(C_i)$, отказов подсистем $P(SEC_i)$ и вероятность системного отказа $P(SE_i)$.

Априорная модель экспериментальной функции «обновление ПО» имеет графико-логическое представление в виде дерева отказов (рис. 4).

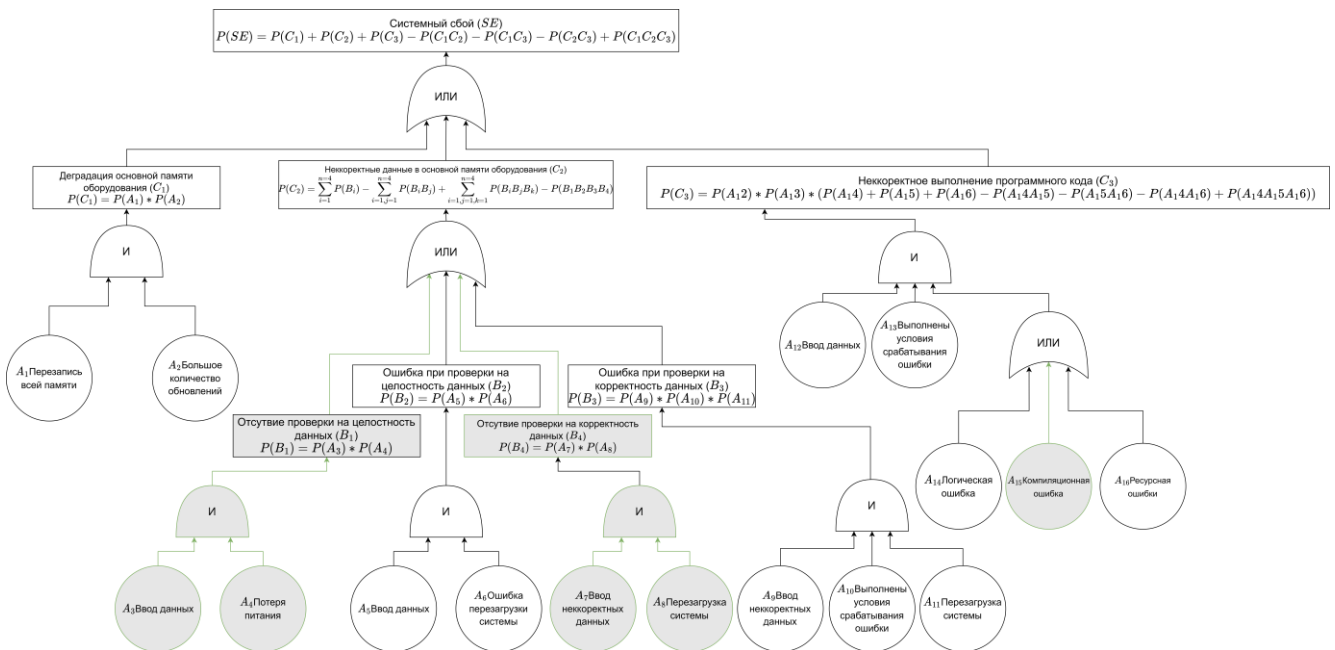


Рисунок 4 – Модель функции «обновление ПО»

На рисунке 4 цветом выделены те события, которые удалось исключить из расчетов посредством доказательства их невероятного наступления за счет анализа исполняемого кода (рис. 5).

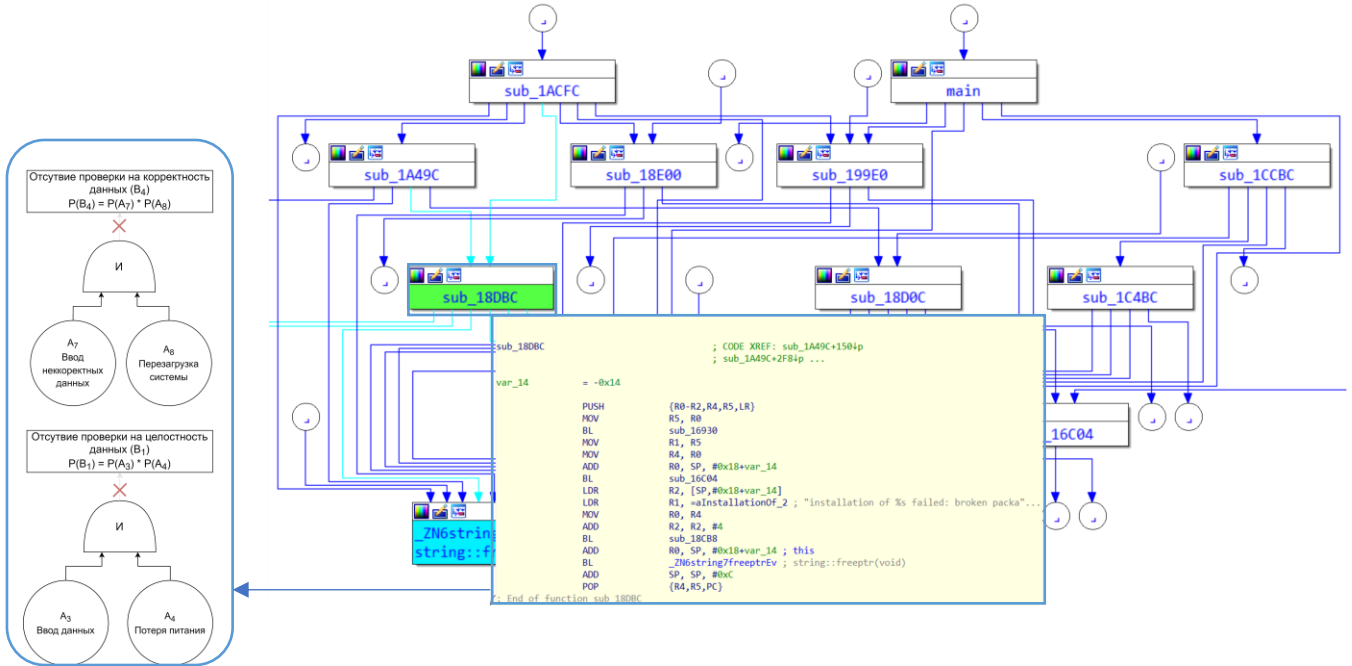


Рисунок 5 – Анализ исполняемого кода функции «обновление ПО»

Тем самым в расчетах не участвуют невероятные отказы, что позволяет сосредоточить усилия на поиске ошибок в тех местах, где это наиболее вероятно.

Для расчётов производительности программная функция представляется в виде конечных состояний S'_i с вероятностями $p_i(t)$. Таким образом выполнение программной функции представляется как марковский процесс с дискретным состоянием и непрерывным временем.

Принимая время перехода системы из состояния S'_i в S'_j как Δt и в предельном случае $\Delta t \rightarrow 0$, получаем систему дифференциальных уравнений Колмогорова (для 4-х состояний) следующего вида:

$$\begin{cases} \dot{p}_0(t) = \lambda_{10}p_1(t) + \lambda_{20}p_2(t) - (\lambda_{01} + \lambda_{02})p_0(t), \\ \dot{p}_1(t) = \lambda_{01}p_0(t) + \lambda_{31}p_3(t) - (\lambda_{10} + \lambda_{13})p_1(t), \\ \dot{p}_2(t) = \lambda_{02}p_0(t) + \lambda_{32}p_3(t) - (\lambda_{20} + \lambda_{23})p_2(t), \\ \dot{p}_3(t) = \lambda_{13}p_1(t) + \lambda_{23}p_2(t) - (\lambda_{31} + \lambda_{32})p_3(t). \end{cases} \quad (27)$$

Для поиска ошибок производительности в программной функции с помощью анализа исполняемого кода строится модель СМО с ограниченной очередью.

Анализ модели позволяет выявлять ошибки, связанные с производительностью, например, места закливания функции. После закливания поток заявок перестает обрабатываться, буфер переполняется и возникают отказы в обслуживании. Таким

образом, выявленная ошибка влияет на производительность функции, которая может отвечать за пересылку пакетов, при этом сама система будет работать без сбоев.

После выявления ошибки в программном компоненте требуется оценить, как найденная ошибка влияет на характеристики качества (этап 11). Для этого разработана методика, которая является развитием методик оценки влияния уязвимостей на систему. Стандартом в этой области стала методика расчета Common Vulnerability Scoring System (CVSS). Усовершенствование этой методики заключается в использовании аппарата нечеткой логики и разработке метрик расчета для функциональной пригодности, надежности и производительности. Для этого разработано 24 метрики, которые объединены в 3 группы. Расчет оценки в каждой группе основан на создании выходных термов, функций принадлежности и базы нечетких правил, которые в свою очередь являются входными переменными для расчета нечеткой оценки влияния ошибки на характеристику качества w_k . В таблице 1 представлены количественные показатели математического метода расчета FE_{bs} .

Таблица 1. Количественные показатели метода оценки влияния ошибки

Название показателя	Количество входных переменных	Количество выходных термов	Количество нечетких правил	Метод дефазификации
Нечеткая оценка влияния на показатели качества (FE_{iss})	9	15	81	Относительно центра области $x_c = \frac{\sum_i \mu(x_i)x_i}{\sum_i \mu(x_i)}$
Нечеткая оценка сложности эксплуатации ошибки (FE_e)	12	18	144	Выбор минимального из максимальных значений
Нечеткая оценка влияния ошибки на компоненты системы (FE_i)	6	15	30	Выбор максимального из максимальных значений
Нечеткая оценка влияния ошибки на характеристики качества (FE_{bs})	9	15	108	Выбор минимального из максимальных значений

Таким образом комплексирование разработанного алгоритма, аналитической модели и метода оценки влияния позволяет реализовать метод оценки показателей

качества компьютерных элементов по исполняемому коду с учетом переноса ПО на ААП (рис. 6).

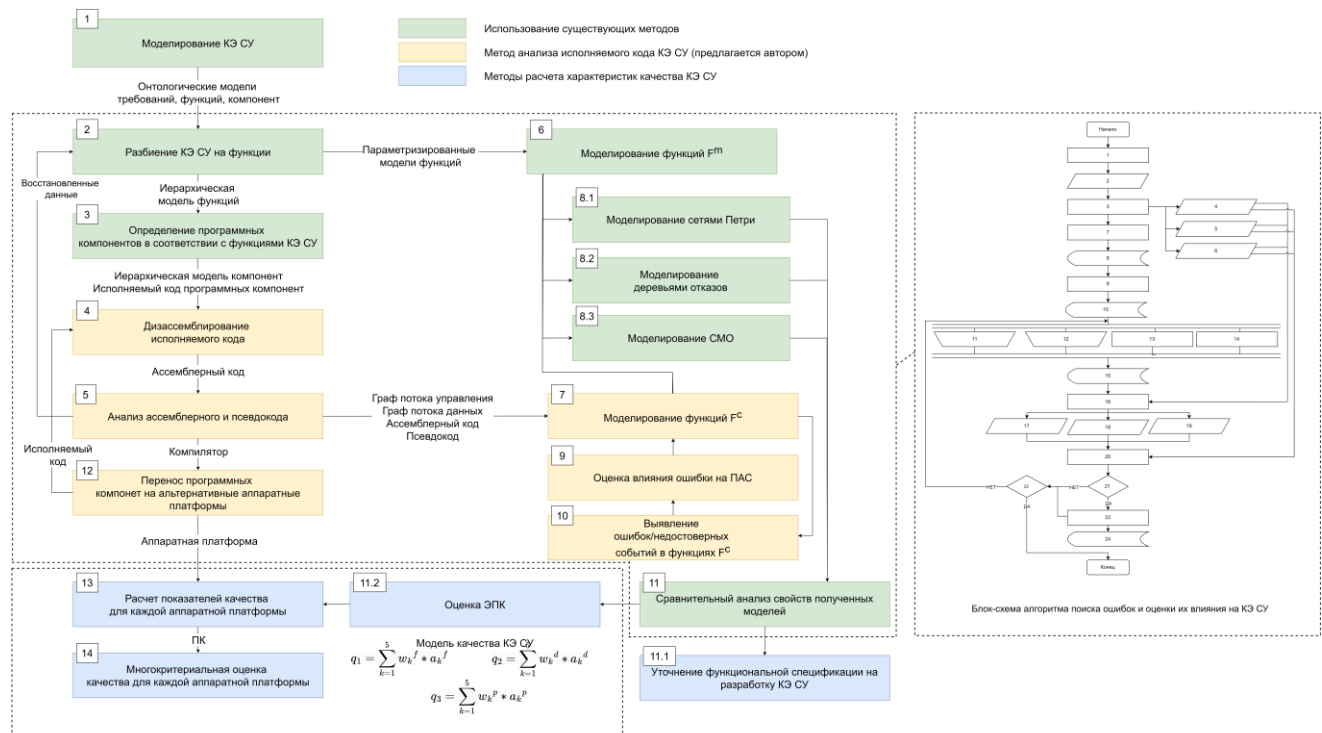


Рисунок 6 – Схема метода оценки качества КЭ СУ

Для определения адекватности предложенного метода была выполнена его проверка в экспериментальной среде.

В четвертой главе проведена экспериментальная проверка разработанной аналитической модели качества КЭ СУ, алгоритма поиска ошибок, метода оценки влияния ошибок на характеристики качества, а также метода оценки качества КЭ СУ при многократном переносе ПО на ААП. Эксперимент проведен для встроенного ПО 126 сетевых устройств, в которых выделено и проанализировано 38 692 программы написанных на языках программирования C/C++ для процессорных архитектур Intel, ARM, MIPS, PowerPC.

Проверка адекватности аналитической модели основана на подтверждении следующих утверждений:

- чем дольше система разрабатывается, тем она качественнее;
- чем сложнее система, тем она менее качественна.

В таблице 2 представлены результаты эксперимента подтверждающего, что ПО, которое разрабатывалось производителем оборудования (проприетарное ПО) в срок до 5 лет, имеет оценки качества ниже, чем у открытого ПО (разрабатывается более 10 лет).

Таблица 2. Результаты анализа проприетарного и открытого ПО

Имя программы	Оценка качества	Примечания
ubnt_spectral.ko	0,419	проприетарный модуль ядра Linux
ubnt_poll.ko	0,438	проприетарный модуль ядра Linux
ubntbox	0,396	проприетарная программа
ubnt_poll_host.ko	0,555	проприетарный модуль ядра Linux
nf_nat_ftp.ko	0,890	модуль ядра Linux
iptables_nat.ko	0,867	модуль ядра Linux
i2cdetect	0.850	открытая программа

На рисунке 7 представлен график, который указывает на то, что аналитическая модель адекватна к утверждению о том, что сложность ПО (метрика Холстеда) влияет на качество.

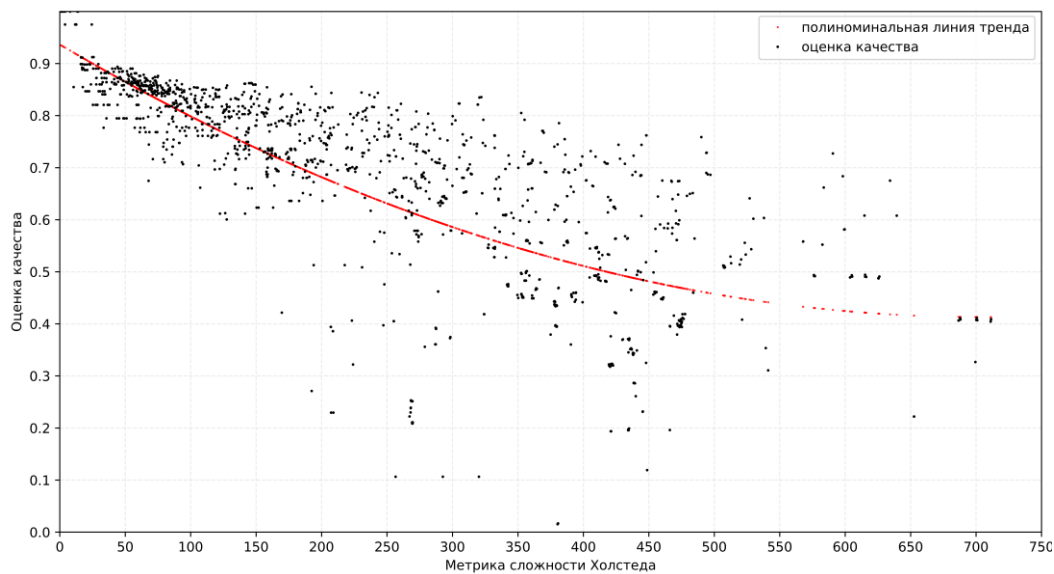


Рисунок 7 – Зависимость качества КЭ СУ от сложности ПО

Таким образом, аналитическая модель качества КЭ СУ адекватно описывает качество реальных объектов.

Экспериментальная проверка алгоритма поиска ошибок показала, что использование анализа исполняемого кода позволяет применять те методы поиска, которые не доступны при переносе ПО на ААП, но доступны при разработке системы по полному ЖЦ.

Пусть в системе имеется e ошибок. Введем следующие показатели:

$P(e)$ – вероятность обнаружения всех ошибок существующими методами поиска

$P(e')$ – вероятность оставшихся ошибок e'

$P(e_s)$ – вероятность обнаружения аппаратно-независимых ошибок

$P(e_h)$ – вероятность обнаружения аппаратно-зависимых ошибок

$S_{11}, S_{12}, S_{13}, S_{14}$ - состояния системы при реализации переноса ПО на ААП, когда исходные коды отсутствуют.

Обозначим $R = \{r_1, \dots, r_n\}$, $Z = \{z_1, \dots, z_n\}$ множество применимых и $M = \{m_1, \dots, m_n\}$, $C = \{c_1, \dots, c_n\}$ неприменимых методов обнаружения аппаратно-зависимых и аппаратно-независимых методов соответственно.

Используя введенные показатели, получаем формулы оценки вероятностей обнаружения ошибок:

$$P(e) = \sum_{r=1}^i P(e_s^r) + \sum_{z=1}^j P(e_h^z) \quad (28)$$

$$P(e') = \sum_{m=1}^i P(e_s^m) + \sum_{m=1}^i P(e_s'^m) + \sum_{r=1}^i P(e_s'^r) + \sum_{c=1}^j P(e_h^c) + \sum_{c=1}^j P(e_h'^c) + \sum_{z=1}^j P(e_h'^z) \quad (29)$$

В таблице 3 представлены результаты аналитических расчетов поиска ошибок без использования анализа исполняемого кода и с его применением, а также выделены цветом те значения, которые удовлетворяют требованиям по вероятности обнаружения ошибок (зеленые), частично не удовлетворяют (желтые), не удовлетворяют (красные) и значения, которые не допустимы (оранжевые).

Таблица 3. Сравнение алгоритмов поиска ошибок при $S_{11}, S_{12}, S_{13}, S_{14}$

Опорное значение $P(e)$	Текущее значение			Применение разработанного алгоритма, $P(e)$
	$P(e_s)$	$P(e_h)$	$P(e)$	
0,9	0,7	0,2	0,78	0,84
0,9	0,6	0,3	0,72	0,82
0,9	0,5	0,4	0,66	0,79
0,9	0,4	0,5	0,6	0,76
0,9	0,3	0,6	0,54	0,74
0,9	0,2	0,7	0,48	0,71
0,9	0	0,9	0,36	0,65

Согласно всем полученным результатам, использование анализа исполняемого кода повышает вероятность обнаружения всех ошибок на 27 %.

Исходя из разработанного алгоритма поиска ошибок, требуется оценить влияние найденных ошибок на систему. Перед применением метода, основанного на аппарате нечеткой логики, проведены сравнительные расчеты базовой оценки CVSS и оценки, рассчитанной на основе нечеткой логики (рис. 8).

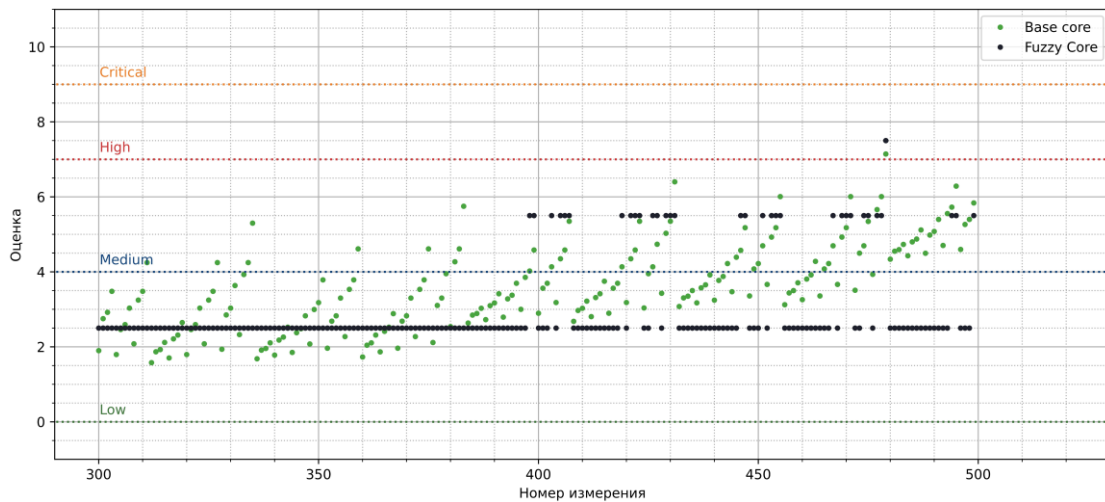


Рисунок 8 – График расчетов оценки CVSS и нечеткой оценки влияния

Оценки совпадают для 75% измерений, что подтверждает адекватность разработанного метода.

Расчёты точности оценки качества по аналогии с методами измерения сводятся к критериям правильность и прецизионность. Правильность методов измерения использует такие понятия как истинное значение или опорное значение. В случаях оценки качества КЭ СУ при переносе ПО на ААП принять истинное значение не представляется возможным. Но, так как значения оценок, полученные при разработке системы по полному ЖЦ, основаны на научных принципах и экспериментальных работах, то следует принять их в качестве опорных значений. Тогда, правильность метода оценки качества КЭ СУ при переносе ПО на ААП определяется систематической погрешностью.

На рисунке 9 представлены оценки качества КЭ СУ, разработанных по полному ЖЦ для архитектуры Intel (рис. 9а) и при переносе ПО на архитектуры ARM, MIPS, PowerPC (рис. 9б).

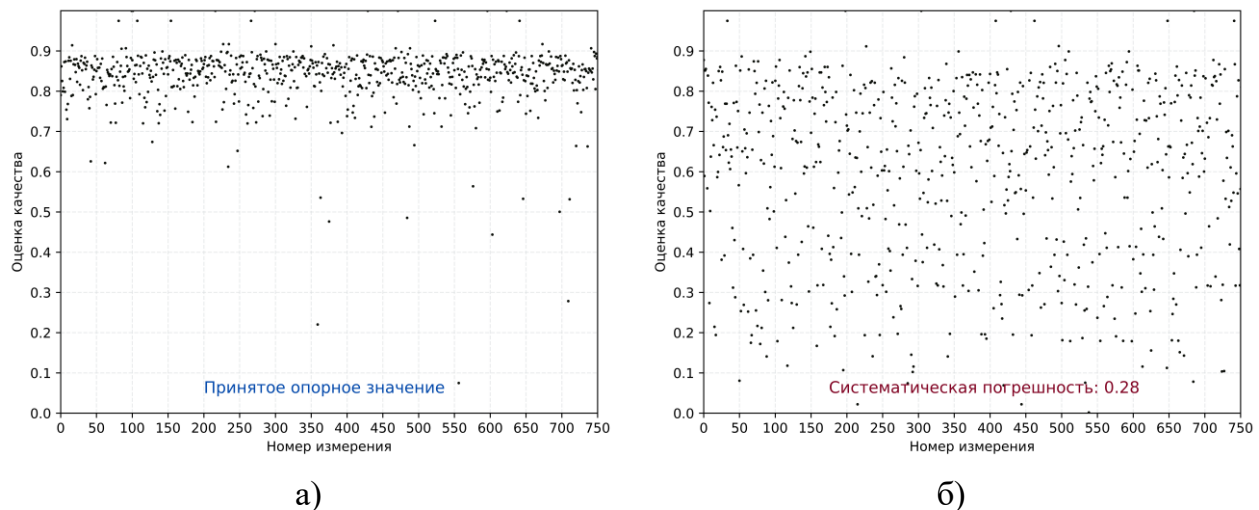


Рисунок 9 – Оценка качества КЭ СУ

На рисунке 10 представлены оценки качества КЭ СУ при переносе ПО на архитектуры ARM, MIPS, PowerPC. Использование разработанной аналитической модели качества (рис. 10а) снижает систематическую погрешность методов оценки качества на 10 %. Так как предлагаемая модель универсальна и не зависит от отсутствия или наличия исходных кодов ПО, то её применение повысит точность оценок при разработке КЭ СУ по полному ЖЦ.

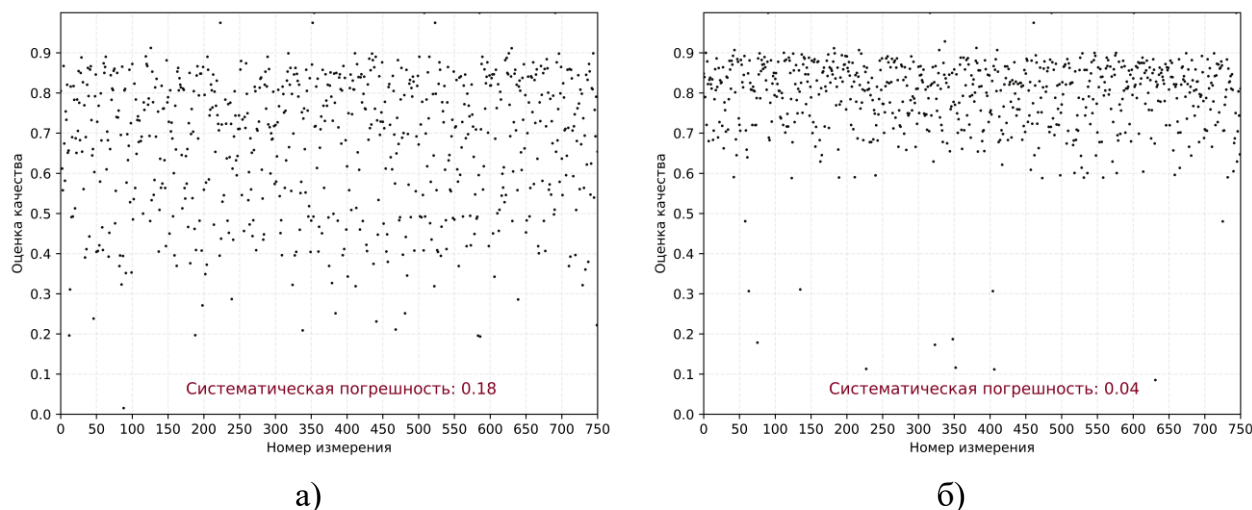


Рисунок 10 – Оценка качества КЭ СУ с применением разработанных методов

Таким образом, использование разработанного метода оценки качества КЭ СУ снижает систематическую погрешность оценки при переносе ПО на ААП на 24 % (рис. 10б).

ЗАКЛЮЧЕНИЕ

Цель научно-квалификационной работы достигнута в результате решения поставленных задач:

1) Проведен анализ обобщенных и аналитических моделей качества вычислительных систем, что позволило выявить проблему моделирования компонент системы для расчета показателей качества в условиях отсутствия исходных кодов. Результаты анализа методов оценки качества свидетельствуют о том, что алгоритмы поиска ошибок в случаях переноса ПО на ААП обнаруживают в среднем на 27 % ошибок меньше. Также анализ показывает, что в процессах оценки учитывается только количество ошибок, но не учитывается качество каждой ошибки. Оба этих фактора повышают систематическую погрешность метода оценки качества.

2) Разработана аналитическая модель качества по трем базовых характеристикам (функциональная пригодность, надежность, производительность). Модель использует 12 ПК, которые в своих расчетах применяют количественные оценки влияния ошибок на характеристики качества.

3) Разработан алгоритм и набор программных инструментов для поиска ошибок и оценки их влияния на характеристики качества. Алгоритм поиска ошибок в исполняемом коде использует существующие методы обратного проектирования, что позволяет обнаружить ошибки в готовом образце системы без информации (техническая документация, исходные коды и др.) требуемой другими методами. Метод оценки влияния ошибки на характеристики качества включает 24 метрики и базу из 363 нечетких правил.

4) Разработан метод оценки качества КЭ СУ, обеспечивающий требуемый уровень точности оценки для эффективного управления процессом модернизации в условиях переноса ПО на ААП, который включает:

- анализ системы с целью выделения значимых функций для оценки качества;
- моделирование программных компонент, которые реализуют выделенные функции системы;
- алгоритм и набор программных инструментов для поиска ошибок в исполняемом коде КЭ СУ;
- метод оценки влияния ошибки на базовые характеристики качества (функциональная пригодность, надежность, производительность);
- аналитическую модель качества КЭ СУ, которая позволяет рассчитать многокритериальную оценку.

5) Разработанный метод оценки качества КЭ СУ прошел экспериментальную проверку для телекоммуникационного оборудования, ПО которого переносится на ААП. Подтверждено, что число найденных ошибок в системе возрастает на 27 %, за счет возникающей возможности обнаружения аппаратно-зависимых ошибок. Установлено, что при использовании разработанного метода правильность метода оценки качества КЭ СУ при переносе ПО на ААП возрастает на 24 %.

Результаты научно-квалификационной работы позволяют снизить систематическую погрешность метода оценки качества КЭ СУ при переносе ПО на ААП, что способствует снижению количества ошибок первого и второго рода в процессе модернизации сложных систем управления.

Разработанные модель и методы могут быть использованы в процессах управления разработкой новых систем по полному ЖЦ для обеспечения качества и безопасности их использования, что позволит снизить систематическую погрешность методов оценивания на 10%. Также исследования позволяют поддерживать требуемой

уровень качества в наследуемых системах, когда техническая информация о них утеряна.

Направление дальнейшего исследования заключается в совершенствовании метода анализа исполняемого кода, за счет применения новых моделей, теории машинного обучения, теории формальной верификации программ. Работа не затрагивает все характеристики качества КЭ СУ, что является точкой роста новых знаний в области системного анализа.

СПИСОК РАБОТ, ОПУБЛИКОВАННЫХ ПО ТЕМЕ ДИССЕРТАЦИИ

Публикации в рецензируемых научных изданиях, включенных в перечень ВАК

1. Нарышкин, К.В. Анализ моделей оценки качества вычислительной системы / К. В. Нарышкин // International Journal of Open Information Technologies. – 2023. – Т. 11, № 10. – С. 44-54.
2. Нарышкин, К.В. Методика оценки влияния программной ошибки на систему с помощью нечеткой логики / К. В. Нарышкин // Автоматизация в промышленности. – 2023. – № 8. – С. 59-64. – DOI 10.25728/avtprom.2023.08.12.
3. Нарышкин, К. В. Модельно-ориентированное проектирование в системе оценки качества телекоммуникационного оборудования / К. В. Нарышкин // International Journal of Open Information Technologies. – 2024. – Т. 12, № 9. – С. 30-39.

Результаты интеллектуальной деятельности

4. Свидетельство о государственной регистрации программы для ЭВМ № 2023669264 Российская Федерация. Программа расчета оценки влияния программной ошибки на вычислительную систему: № 2023668181: заявл. 28.08.2023: опубл. 12.09.2023 / К. В. Нарышкин; заявитель и правообладатель Нарышкин Константин Викторович.

Публикации в других изданиях и сборниках научных трудов и конференций

5. Нарышкин, К.В. Кластеризация отчетов об исправлениях в программном обеспечении сетевого оборудования / К. В. Нарышкин // Вестник Концерна ВКО "Алмаз – Антей". – 2023. – № 2. – С. 90-100. – DOI 10.38013/2542-0542-2023-2-90-100.
6. Нарышкин К.В. Инструменты сравнения бинарного кода / К. В. Нарышкин // Системный администратор. – 2023. – № 12(253). – С. 90-94.
7. Нарышкин, К.В. Факторы возникновения неконтролируемого поведения информационных систем при трансфере технологий / К. В. Нарышкин // Сборник

докладов научно-практической конференции «Цифровые технологии и их приложения», ЦТМ ФГБОУ ВО «УГНТУ», Уфа, 2021. [Электронная версия].

8. Нарышкин, К.В. Описательная модель системы обратного проектирования программного обеспечения / К. В. Нарышкин // Приоритетные направления инновационной деятельности в промышленности : сборник научных статей V международной научной конференции в 2-х частях, Казань, 30–31 мая 2021 года / НПП МЕДПРОМДЕТАЛЬ ООО Газпром трансгаз Казань. Том Часть 2. – Казань: Общество с ограниченной ответственностью "КОНВЕРТ", 2021. – С. 51-53.

9. Нарышкин, К. В. Анализ открытой информации об исправлениях в программном обеспечении встраиваемых систем / К. В. Нарышкин // Актуальные проблемы прикладной информатики в образовании, экономике, государственном и муниципальном управлении: Материалы Международной научной конференции, Барнаул, 25 мая 2022 года / Под редакцией А.Ю. Юдинцева, Г.Н. Трошкиной. Том Выпуск VII. – Барнаул: Алтайский государственный университет, 2022. – С. 72-76.

10. Нарышкин, К. В. Анализ отчетов изменений в программном обеспечении встраиваемых систем / К. В. Нарышкин, В. А. Новиков // Актуальные тренды цифровой трансформации промышленных предприятий: сборник статей Всероссийской научно-практической конференции, Казань, 21–24 сентября 2022 года. – Курск: Закрытое акционерное общество "Университетская книга", 2022. – С. 204-208.

11. Нарышкин, К.В. Модель качества иностранной информационной системы / К. В. Нарышкин // Российская наука в современном мире: Сборник статей XLVIII международной научно-практической конференции, Москва, 31 августа 2022 года. – Москва: Общество с ограниченной ответственностью "Актуальность.РФ", 2022. – С. 25-28.

12. Нарышкин, К.В. Кластерный анализ открытых отчетов об ошибках программного обеспечения встраиваемых систем / К. В. Нарышкин // Сборник тезисов докладов VII научно-технической конференции «Математическое моделирование, инженерные расчеты, и программное обеспечение для решения задач ВКО», Москва, 2022. - с.44.

13. Нарышкин, К. В. Алгоритм подготовки к инструментации встроеного программного обеспечения / С. В. Карбовский, К. В. Нарышкин // Кибербезопасность: технические и правовые аспекты защиты информации : Сборник научных трудов I Национальной научно-практической конференции, Москва, 24–26 мая 2023 года. – Москва: МИРЭА - Российский технологический университет, 2023. – С. 34-37.

14. Нарышкин, К.В. Обратное проектирование как метод системного анализа бинарного кода / К. В. Нарышкин // Сборник тезисов докладов VIII научно-технической конференции «Математическое моделирование, инженерные расчеты, и программное обеспечение для решения задач ВКО», Москва, 2023. - с.32-33.

15. Нарышкин, К. В. Метод анализа пространства состояний программы при поиске уязвимостей в исполняемом коде / К. В. Нарышкин // Кибербезопасность: технические и правовые аспекты защиты информации: Сборник научных трудов III Национальной научно-практической конференции, Москва, 23–24 апреля 2024 года. – Москва: МИРЭА - Российский технологический университет, 2024. – С. 18-23.

Подписано в печать 30.06.2025. Формат 60×84 1/16.
1,0 а.л. Тираж 100 экз.

Отпечатано в СПбГУТ, 193232, Санкт-Петербург, пр. Большевиков, д. 22, корп. 1