

**УДК 004.05**

**Ломазов Вадим Александрович**, доцент кафедры информатики и информационных технологий, Белгородский государственный национальный исследовательский университет, г. Белгород

**Пономарев Денис Сергеевич**, аспирант, Белгородский государственный национальный исследовательский университет, г. Белгород

**МОДЕЛИ И АЛГОРИТМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В  
ЗАДАЧАХ УПРАВЛЕНИЯ КАЧЕСТВОМ ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ**

**Аннотация**

В современных условиях цифровизации и усложнения программных систем вопросы обеспечения и управления качеством программного обеспечения приобретают особую значимость. Традиционные методы контроля качества, основанные преимущественно на экспертных оценках и ручных процедурах тестирования, не в полной мере отвечают требованиям масштабируемости, воспроизводимости и предсказуемости результатов. В статье рассматриваются современные модели и алгоритмы поддержки принятия решений в задачах управления качеством программного обеспечения. Особое внимание уделяется формализованным моделям качества, ориентированным на операционализацию абстрактных характеристик, а также алгоритмическим подходам, основанным на вероятностных методах и машинном обучении. Проанализированы возможности применения подхода Quamoco и activity-based quality models в сочетании с байесовскими сетями для оценки и прогнозирования качества программных систем. Рассматривается роль алгоритмов машинного обучения в автоматизации процессов управления качеством. Обоснована целесообразность комплексного подхода, объединяющего формальные

модели и методы интеллектуального анализа данных для построения систем поддержки принятия решений в области QA.

### **Annotation**

In modern conditions of digitalization and increasing complexity of software systems, issues of software quality assurance and management are becoming particularly important. Traditional quality control methods, based primarily on expert assessments and manual testing procedures, do not fully meet the requirements of scalability, reproducibility, and predictability of results. The article discusses modern models and algorithms for decision support in software quality management tasks. Special attention is paid to formalized quality models focused on the operationalization of abstract characteristics, as well as algorithmic approaches based on probabilistic methods and machine learning. The possibilities of using the Quamoco approach and activity-based quality models in combination with Bayesian networks for evaluating and predicting the quality of software systems are analyzed. The role of machine learning algorithms in the automation of quality management processes is considered. The expediency of an integrated approach combining formal models and methods of data mining for building decision support systems in the field of QA is substantiated.

**Ключевые слова:** качество программного обеспечения, управление качеством, модели качества, поддержка принятия решений, Quamoco, байесовские сети, машинное обучение, прогнозирование дефектов.

**Keywords:** software quality, quality management, quality models, decision support, Quamoco, Bayesian networks, machine learning, defect prediction.

Рост сложности программных систем, увеличение объёмов исходного кода, активное внедрение распределённых архитектур и микросервисных решений привели к существенному усложнению задач управления качеством программного обеспечения. Качество перестало рассматриваться исключительно как итоговый результат тестирования на завершающем этапе разработки. В современных условиях оно выступает как непрерывный

управляемый процесс, охватывающий все стадии жизненного цикла программного продукта. При этом требования к надёжности, безопасности, сопровождаемости и производительности программных систем неуклонно возрастают, что делает традиционные подходы обеспечения качества всё менее эффективными [1].

Практика показывает, что опора исключительно на ручное тестирование и субъективные экспертные оценки не позволяет обеспечить требуемый уровень воспроизводимости и обоснованности управленческих решений. Особенно это проявляется в условиях ограниченных ресурсов, жёстких сроков разработки и необходимости частых релизов. В такой ситуации возрастает потребность в формализованных моделях, позволяющих количественно описывать качество программного обеспечения, а также в алгоритмах, обеспечивающих поддержку принятия решений на основе объективных данных.

Современные исследования в области инженерии программного обеспечения направлены на разработку операционализированных моделей качества, связывающих абстрактные характеристики программных систем с измеряемыми параметрами, а также на создание интеллектуальных алгоритмов анализа данных, способных прогнозировать дефектность и поддерживать управление качеством.

Формализация качества программного обеспечения и проблемы классических моделей. Качество программного обеспечения традиционно рассматривается как совокупность свойств, определяющих его способность удовлетворять потребности пользователей и заинтересованных сторон. К числу таких свойств обычно относят надёжность, функциональную полноту, удобство использования, производительность, сопровождаемость и ряд других характеристик [2].

Классические модели качества, оперирующие перечнем атрибутов или «ility» характеристик, не предоставляют однозначных механизмов перехода от понятийного уровня к уровню конкретных метрик [3]. В результате измерения

качества часто носят фрагментарный характер, а интерпретация результатов зависит от субъективных представлений специалистов. Это снижает объективность оценки и затрудняет сопоставление различных программных продуктов или версий одного и того же продукта.

Особую проблему представляет агрегирование отдельных метрик в интегральную оценку качества. В отсутствие формализованных правил агрегирования возникает риск логических противоречий, когда улучшение отдельных показателей не приводит к росту общей оценки или даже сопровождается её снижением. В этих условиях актуализируется задача построения таких моделей качества, которые обеспечивают строгую логическую связь между измеряемыми параметрами и абстрактными характеристиками программного обеспечения [4].

Одним из наиболее известных современных подходов к формализации качества программного обеспечения является модель Quamoso. Её основная идея заключается в создании метамодели, позволяющей связать высокоуровневые характеристики качества с конкретными измеряемыми свойствами программного продукта. В рамках Quamoso вводится понятие «product factor», которое отражает определённое свойство программной системы, влияющее на качество и поддающееся измерению с использованием инструментальных средств [5].

Такая структура обеспечивает переход от абстрактных категорий, например сопровождаемости или надёжности, к конкретным метрикам, получаемым средствами статического и динамического анализа. Базовая модель Quamoso является независимой от конкретных технологий, что позволяет адаптировать её под различные доменные области за счёт модульного расширения. На практике данная модель была детально реализована для языков программирования Java и C++, где была сформирована обширная система факторов и измерений, обеспечивающая автоматизированную оценку свойств программных систем [6,7].

Эффективность её применения зависит от полноты набора метрик, корректности их интерпретации и адекватности используемых весовых коэффициентов. Кроме того, модель в большей степени ориентирована на статические характеристики программного кода и в меньшей степени учитывает динамические аспекты работы системы и особенности процесса разработки.

Activity-based quality models и вероятностный подход к оценке качества. Для преодоления ограничений сугубо статических моделей качества были предложены «activity-based quality models», в которых качество программного обеспечения интерпретируется через влияние свойств системы на деятельность пользователей и сопровождающих специалистов. В таких моделях основное внимание уделяется тому, каким образом характеристики программного продукта воздействуют на процессы его эксплуатации, сопровождения и развития [8].

Преобразование activity-based модели в байесовскую сеть позволяет формализовать причинно-следственные связи между характеристиками программного обеспечения и вероятностными показателями дефектности и деградации качества. Это обеспечивает возможность учитывать неопределённость, присущую как измерениям, так и динамике процессов разработки.

Байесовская сеть представляет собой вероятностную модель, в узлах которой размещаются случайные величины, соответствующие характеристикам системы, а рёбра отражают зависимости между ними. После задания структуры сети и параметров распределений становится возможным проводить вычисления апостериорных вероятностей, что позволяет оценивать текущее состояние качества и строить прогнозы относительно его изменения при варьировании входных параметров [8].

Эмпирические исследования, проведённые на данных различных программных проектов, подтверждают применимость данного подхода для практических задач управления качеством. Байесовские сети позволяют

оценивать вероятность возникновения дефектов, прогнозировать влияние изменений в коде или процессе разработки на надёжность системы, а также учитывать неопределённость измерений. Вместе с тем построение таких моделей требует значительных усилий по сбору и подготовке данных, а также по калибровке вероятностных зависимостей, что ограничивает их широкое промышленное применение [9].

Алгоритмы машинного обучения в задачах управления качеством программного обеспечения. В последние годы всё более широкое распространение получают методы машинного обучения, ориентированные на решение задач обеспечения качества программного обеспечения. Наиболее активно такие методы применяются для прогнозирования дефектов, автоматизации тестирования и приоритизации тестовых сценариев. Использование обучаемых моделей позволяет обрабатывать большие объёмы данных, формируемых в процессе разработки, и выявлять скрытые закономерности, недоступные традиционному статистическому анализу [10].

В качестве обучающих признаков в моделях машинного обучения используются метрики кода, параметры изменений в системе контроля версий, данные о результатах тестирования, характеристики разработчиков и другие показатели. На основании этих данных формируется модель, способная с определённой вероятностью предсказывать дефектность отдельных модулей программной системы или эффективность различных стратегий тестирования.

Практика применения машинного обучения демонстрирует, что при наличии репрезентативных обучающих выборок такие модели способны существенно повысить эффективность процессов QA. Это проявляется в более точной локализации уязвимых компонентов системы, снижении объёмов избыточного тестирования и более рациональном распределении ресурсов. Вместе с тем устойчивость результатов машинного обучения во многом зависит от качества исходных данных и их соответствия текущему состоянию проекта [11].

К числу существенных ограничений машинного обучения следует отнести проблему интерпретируемости результатов. Во многих случаях, особенно при использовании глубоких нейронных сетей, становится затруднительно объяснить, по каким причинам модель приняла то или иное решение. Это создаёт определённые сложности при использовании таких алгоритмов в управленческих процессах, где требуется обоснование принимаемых решений и их согласование с экспертными оценками.

Интеграция моделей качества и алгоритмов для поддержки принятия решений. Анализ современных подходов позволяет сделать вывод о целесообразности интеграции формальных моделей качества и алгоритмических методов анализа данных при создании систем поддержки принятия решений в области управления качеством программного обеспечения. Формальная модель качества задаёт структуру представления характеристик программного продукта и определяет, какие параметры подлежат измерению и контролю. Алгоритмический уровень обеспечивает обработку собранных данных, учёт неопределённости и построение прогнозных оценок [12].

В рамках такого интегрированного подхода формируется замкнутый контур управления качеством, в котором данные о состоянии программного продукта непрерывно обновляются, интерпретируются с использованием вероятностных и обучаемых моделей и используются для выработки управленческих решений. Это позволяет перейти от реактивного устранения дефектов к проактивному управлению качеством, ориентированному на предупреждение проблем и оптимизацию процессов разработки.

Важным аспектом интеграции является необходимость согласования результатов алгоритмической обработки с экспертными знаниями. Формальные модели и машинное обучение не могут полностью заменить профессиональные суждения специалистов, однако способны существенно повысить их обоснованность и снизить долю субъективизма. В этом контексте системы поддержки принятия решений следует рассматривать как инструмент

усиления аналитических возможностей специалистов, а не как полностью автономные механизмы управления качеством.

Ограничения и перспективы развития интеллектуальных систем управления качеством. Несмотря на очевидные преимущества использования формализованных моделей и интеллектуальных алгоритмов, на пути их внедрения в промышленную практику существуют существенные ограничения. Одним из ключевых факторов является зависимость эффективности таких систем от качества исходных данных. В условиях отсутствия достаточной истории изменений, тестирования и эксплуатации программных систем применение сложных алгоритмов может не дать ожидаемого эффекта.

Формальные модели, такие как Quamoco и activity-based quality models, обеспечивают теоретическую основу для объективной оценки качества и перехода от абстрактных характеристик к измеримым параметрам. Вероятностные методы и алгоритмы машинного обучения расширяют возможности таких моделей за счёт учёта неопределённости, автоматизации анализа и построения прогнозных оценок.

Совместное применение формализованных моделей качества и интеллектуальных алгоритмов создаёт предпосылки для формирования эффективных систем поддержки принятия решений в области QA. Такие системы позволяют повысить управляемость процессов обеспечения качества, снизить издержки на тестирование и сопровождение, а также обеспечить более высокий уровень надёжности программных продуктов. Вместе с тем дальнейшее развитие данного направления требует углубления эмпирических исследований, совершенствования интерпретируемости алгоритмов и расширения практики их внедрения в реальные проекты.

## Литература

1. Черняховская Л. Р., Малахова А. И. Разработка моделей и методов интеллектуальной поддержки принятия решений на основе онтологии

организационного управления программными проектами // Онтология проектирования. 2013. №4 (10). URL: <https://cyberleninka.ru/article/n/razrabotka-modeley-i-metodov-intellektualnoy-podderzhki-prinyatiya-resheniy-na-osnove-ontologii-organizatsionnogo-upravleniya> (дата обращения: 09.12.2025).

2. Томилов, А. В. Разработка веб-приложения для системы поддержки принятия решения руководителя ИТ-подразделения промышленного предприятия / А. В. Томилов. — Текст : непосредственный // Молодой ученый. — 2025. — № 34 (585). — С. 1-11.
3. Wagner S., Goeb A., Heinemann L., Kläs M., Lampasona C., Lochmann K., Mayr A., Plösch R., Seidl A., Streit J., Trendowicz A. Operationalised product quality models and assessment: The Quamoco approach // Information and Software Technology. 2015. Vol. 62. P. 101–123.
4. Wagner S. A Bayesian network approach to assess and predict software quality using activity-based quality models // Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE). Miami, USA, 2011. P. 367–372.
5. Wagner S. Software product quality control. Springer, 2013. 309 p.
6. Basciani F., Di Pompeo D., Di Rocco J., Pierantonio A. A customizable approach to assess software quality through multi-criteria decision making // Journal of Systems and Software. 2023. Vol. 197. P. 1–18.
7. Omri M., Sinz C. Machine learning techniques for software quality assurance: A survey // IEEE Access. 2021. Vol. 9. P. 121140–121160.
8. Laracy J. Software quality assurance and artificial intelligence: A systems-theoretic perspective // Systems. 2025. Vol. 4, No. 4. P. 1–21.
9. Chang Y. H., Horng S. J., Lin R. J. A quantity model for controlling and measuring software quality based on the expert decision-making algorithm // Journal of Software Engineering and Applications. 2009. Vol. 2. P. 1–9.

10. Fenton N. E., Neil M. Software metrics: Roadmap // Proceedings of the Conference on the Future of Software Engineering. Limerick, Ireland, 2000. P. 357–370.
11. Kitchenham B., Pfleeger S. L. Software quality: The elusive target // IEEE Software. 1996. Vol. 13, No. 1. P. 12–21.
12. Basili V. R., Caldiera G., Rombach H. D. The Goal Question Metric approach // Encyclopedia of Software Engineering. New York: Wiley, 1994. P. 528–532.

### **Literature**

1. Chernyakhovskaya L. R., Malakhova A. I. Development of models and methods of intellectual decision support based on the ontology of organizational management of software projects // The ontology of design. 2013. No. 4 (10). URL: <https://cyberleninka.ru/article/n/razrabotka-modeley-i-metodov-intellektualnoy-podderzhki-prinyatiya-resheniy-na-osnove-ontologii-organizatsionnogo-upravleniya> (date of request: 09.12.2025).
2. Tomilov, A.V. Development of a web application for a decision support system for the head of the IT department of an industrial enterprise / A.V. Tomilov. — Text : direct // Young scientist. — 2025. — № 34 (585). — Pp. 1-11.
3. Wagner S., Geb A., Heinemann L., Klas M., Lampasona S., Lohmann K., Mayr A., Plesh R., Seidl A., Streit J., Trendovich A. Operational models and product quality assessment.: The Quamoco approach // Information and software technologies. 2015. Volume 62. pp. 101-123.
4. Wagner S. Bayesian network approach to software quality assessment and forecasting using action-based quality models // Proceedings of the 23rd International Conference on Software Development and Knowledge Engineering (SEKE). Miami, USA, 2011. pp. 367-372.
5. Wagner S. Quality control of software products. Springer, 2013. 309 p.
6. Basciani F., Di Pompeo D., Di Rocco J., Pierantonio A. Customizable approach to software quality assessment through multi-criteria decision making // Journal of Systems and Software. 2023. Volume 197. pp. 1-18.

7. Omri M., Sints S. Machine learning methods for software quality assurance: an overview // IEEE Access. 2021. Volume 9. pp. 121140-121160.
8. Larasi J. Software quality assurance and artificial intelligence: a system-theoretical perspective // Systems. 2025. Volume 4, No. 4. pp. 1-21.
9. Chang Y. H., Horng S. J., Lin R. J. A quantitative model for software quality control and measurement based on an expert decision-making algorithm // Journal of Software Engineering and Applications. 2009. Volume II. 2. pp. 1-9.
10. Fenton N. E., Neil M. Program indicators: a roadmap // Materials of the conference on the future of software engineering. Limerick, Ireland, 2000. pp. 357-370.
11. Kitchenham B., Pfleger S. L. Software quality: an unattainable goal // IEEE Software. 1996. Volume 13, No. 1. pp. 12-21.
12. Bazili V. R., Kaldiera G., Rombach H. D. A metric approach to goal definition // Encyclopedia of Software Engineering. New York: Wiley, 1994. pp. 528-532.