

**ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES****DOI:** <https://doi.org/10.60797/IRJ.2026.164.17>**ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРИЛОЖЕНИЯ ОБУЧАЮЩИХ ТРЕХМЕРНЫХ МОДЕЛЕЙ ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ**

Научная статья

**Левина Т.М.<sup>1</sup>, Шамсутдинова А.В.<sup>2,\*</sup>, Шаймарданов И.Ф.<sup>3</sup>**<sup>1</sup> ORCID : 0000-0002-5959-6692;<sup>1, 2, 3</sup> Институт нефтепереработки и нефтехимии Уфимского государственного нефтяного технического университета, Салават, Российская Федерация

\* Корреспондирующий автор (shamsutdinova.azalya[at]yandex.ru)

**Аннотация**

В статье обосновывается необходимость функционального тестирования программных приложений с обучающими трехмерными моделями технологического оборудования. В современных условиях стремительного развития цифровых технологий особое значение приобретает качество программного обеспечения. Ошибки и недоработки программных продуктов могут привести к потере данных и финансовым убыткам, следовательно, тестирование является одним из важнейших этапов жизненного цикла разработки программных обеспечений. Оно позволяет выявить несоответствия между фактическими и требуемыми характеристиками программного продукта, исключить возникновение критических инцидентов в процессе обучения. Для обучающих систем, имитирующих работу сложного технологического оборудования, этап тестирования приобретает стратегическое значение.

**Ключевые слова:** функциональное тестирование, программное приложение, трехмерные модели, технологическое оборудование, метод черного ящика, сценарий тестирования, обучающие системы, нефтегазовое оборудование.

**FUNCTIONAL TESTING OF SOFTWARE APPLICATIONS FOR TRAINING THREE-DIMENSIONAL MODELS OF TECHNOLOGICAL EQUIPMENT**

Research article

**Levina T.M.<sup>1</sup>, Shamsutdinova A.V.<sup>2,\*</sup>, Shaimardanov I.F.<sup>3</sup>**<sup>1</sup> ORCID : 0000-0002-5959-6692;<sup>1, 2, 3</sup> Institute of Oil Refining and Petrochemistry of Ufa State Petroleum Technical University, Salavat, Russian Federation

\* Corresponding author (shamsutdinova.azalya[at]yandex.ru)

**Abstract**

The article substantiates the need for functional testing of software applications with training three-dimensional models of technological equipment. In today's rapidly developing digital technology environment, software quality is of particular importance. Errors and flaws in software products can lead to data loss and financial losses, therefore, testing is one of the most important stages in the software development life cycle. It allows to identify discrepancies between the actual and required characteristics of a software product and prevent critical incidents from occurring during the training process. For training systems that simulate the operation of complex technological equipment, the testing stage is of strategic importance.

**Keywords:** functional testing, software application, three-dimensional models, technological equipment, black box method, test scenario, training systems, oil and gas equipment.

**Введение**

Программное приложение обучающих трехмерных моделей технологического оборудования требует не только отображение полигональных моделей, но и гарантии корректной работы всех его функциональных компонентов [1]. Гарантия корректной работы всех функциональных компонентов, таких как вращение модели, выбор элемента и другие взаимодействия с графическим интерфейсом программы, позволит осваивать оборудования и надежно эксплуатировать его в дальнейшем. Функциональное тестирование становится ключевым этапом жизненного цикла перед его внедрением в промышленную эксплуатацию.

Актуальность исследования обусловлена стремительным распространением трехмерного моделирования в области профессионального обучения, особенно в нефтегазовом секторе. Качественные 3D-модели, точно отражающие реальное оборудование, стали основой эффективных VR-тренажеров [2]. Однако, даже визуально совершенная модель теряет ценность, если её интерактивные функции работают некорректно. Ошибки в логике взаимодействия, неверная обработка пользовательских действий или сбои в отображении элементов могут привести к формированию у обучающихся неверных навыков, что в реальных условиях чревато авариями и финансовыми потерями. В связи с этим системное функциональное тестирование, ориентированное на проверку именно пользовательских сценариев, становится критически важным.

Целью данного исследования является разработка практической методики функционального тестирования программного приложения с обучающими трехмерными моделями технологического оборудования на основе метода «черного ящика».

Задачи исследования:

- провести анализ современных методов и инструментов функционального тестирования, в особенности для приложений с трехмерной графикой.
- разработать детальный сценарий функционального тестирования для типового блока модели «Емкость», охватывающий ключевые пользовательские операции.
- апробировать разработанный сценарий на примере конкретного программного приложения.
- предложить классификацию выявленных дефектов по степени их критичности для учебного процесса.
- сформулировать рекомендации по внедрению методики в цикл разработки аналогичных обучающих систем.

### **Теоретические аспекты функционального тестирования**

В соответствии с ГОСТ Р 56939-2024 функциональное тестирование является видом работ по исследованию программы, направленный на выявление отличий между ее реально существующими и требуемыми свойствами, который решает сразу несколько критически важных задач:

- проверяет корректность работы всех сценариев взаимодействия с трехмерными моделями;
- выявляет ошибки в логике обучающего модуля;
- обеспечивает стабильность интерфейса и его реакцию на пользовательские действия;
- помогает разработчикам и методистам поддерживать соответствие между теоретической моделью и ее цифровой реализацией;
- определяет, выполняются ли требования безопасности, идентифицированные в процессе анализа требований к ПО [3].

Рассмотрим основные типы функциональных тестов (рис. 1).



Рисунок 1 - Типы функциональных тестов

DOI: <https://doi.org/10.60797/IRJ.2026.164.17.1>

Из рисунка 1 видно, что UX-тестирование является ключевым приоритетом, набрав 28%. Это означает, что в 2025 году бизнес осознает, что даже безупречно работающая с технической точки зрения система бесполезна, если она неудобна для пользователя. Компании направляют значительные ресурсы на проверку удобства, интуитивности и эмоционального отклика продукта, понимая, что именно качество пользовательского опыта становится главным конкурентным преимуществом на насыщенном рынке [4]. Лидирующая позиция функционального тестирования наглядно показывает, что сегодня недостаточно просто создать работающий продукт — он должен нравиться пользователям, поскольку именно от их удовлетворенности напрямую зависит коммерческий успех.

### **Методы функционального тестирования**

Для выполнения функционального тестирования применяются современные методики испытаний, которые объединяют ручные и автоматизированные подходы, включая использование аналитики для оптимизации тестовых сценариев [5].

Рассмотрим основные методики испытаний, классифицированные по принципам доступа к внутренней структуре программы и характеру тестируемых функций:

- метод «Черного ящика». Целью данного метода заключается в том, чтобы убедиться, что программа выполняет заявленные функции корректно. А также тестировщик не имеет доступа к исходному коду. Программные средства для проведения данного метода: Selenium, Appium, Katalon Studio, TestComplete.

- метод «Белого ящика». Цель — проверить логику кода, ветвления, циклы, обработку исключений. Тестировщик анализирует внутреннюю структуру и алгоритмы программы, применяется программистами и автоматизированными системами. Для данного метода используются следующие программы — JUnit, NUnit, PyTest, TestNG.

- метод «Серого ящика». Комбинирует принципы «черного» и «белого» ящика, включает в себя частичное знание внутренней структуры программы, часто используется при тестировании API и интеграционных сценариев. Используются такие программы, как Postman, SoapUI, ReadyAPI, Robot Framework.

Одним из часто используемых методов является метод «Черного ящика». Данный метод был выбран благодаря своей универсальности и ориентации на конечного пользователя, что позволяет проверять функциональность приложения именно так, как это будет делать оператор установки, не углубляясь в детали реализации кода [8].

#### Ключевые техники тестирования методом «черного ящика»:

- эквивалентное разбиение — разделение входных данных на классы, обрабатываемые программой одинаково, что сокращает количество тестов. Пример: для диапазона от А до В выделяются валидные и невалидные классы значений.
- анализ граничных значений — тестирование значений на границах диапазонов и за их пределами, где наиболее вероятны ошибки. Пример: для диапазона  $-99 \leq N \leq 99$  проверяются  $\min-1$ ,  $\min$ ,  $\max$ ,  $\max+1$ .
- тестирование таблицы решений — проверка бизнес-логики через таблицу условий и действий, где каждый столбец представляет уникальное бизнес-правило и тестовый сценарий.
- тестирование переходов состояний — проверка корректных и некорректных переходов между состояниями системы, применяется для систем с четкими состояниями, например, банкоматы [9].

#### Составление сценария тестирования

С учетом выше изложенного составим сценарий функционального тестирования трехмерных моделей программного приложения (табл. 1) [6], [7].

Таблица 1 - Сценарий тестирования блока модели «Емкость»

DOI: <https://doi.org/10.60797/IRJ.2026.164.17.2>

| № | Текущее состояние                     | Событие  | Новое состояние   | Ожидаемый результат   |
|---|---------------------------------------|--|-------------------|---|
| 1 | Главное меню                          | Выбрать «Емкость»                                | Интерфейс емкости | Отображение трехмерной модели емкости с надписями элементов в левой части экрана, SCADA-схемы в правой части экрана и поля для ввода значения температуры в левой верхней части экрана. |
| 2 | Интерфейс емкости (модель неподвижна) | Нажать стрелку «Вправо»                          | Вращение вправо   | Трехмерная модель начинает вращаться вправо   |
| 3 | Интерфейс емкости (модель неподвижна) | Нажать стрелку «Влево»                           | Вращение влево    | Трехмерная модель начинает вращаться влево  |
| 4 | Вращение вправо                       | Нажать стрелку «Вправо»                          | Модель неподвижна | Трехмерная модель останавливается   |
| 5 | Вращение вправо                       | Нажать стрелку «Влево»                           | Вращение влево    | Трехмерная модель мгновенно меняет направление и вращается влево  |
| 6 | Вращение влево                        | Нажать стрелку «Влево»                           | Модель неподвижна | Трехмерная модель останавливается   |
| 7 | Вращение влево                        | Нажать стрелку «Вправо»                          | Вращение вправо   | Трехмерная модель мгновенно меняет направление и вращается вправо   |
| 8 | Интерфейс емкости                     | Ввести корректное значение температуры — 160–170 | Интерфейс емкости | Под SCADA-схемой отображается сообщение о правильном вводе значения температуры   |
| 9 | Интерфейс емкости                     | Ввести повышенное                                | Интерфейс емкости | Под SCADA-схемой  |

| №  | Текущее состояние | Событие  | Новое состояние   | Ожидаемый результат  |
|----|-------------------|--|-------------------|--|
|    |                   | значение температуры — 171–190   |                   | отображается сообщение о превышенном значении температуры                            |
| 10 | Интерфейс емкости | Ввести пониженное значение температуры — 0–160   | Интерфейс емкости | Под SCADA-схемой отображается сообщение о пониженном значении температуры            |
| 11 | Интерфейс емкости | Ввести критически повышенное значение температуры — >190                                     | Интерфейс емкости | Под SCADA-схемой отображается сообщение о критически повышенном значении температуры |
| 12 | Интерфейс емкости | Ввести любую недопустимое значение в поле для ввода (отрицательное значение, символы, буквы) | Интерфейс емкости | Под SCADA-схемой отображается сообщение неверно введенном формате                    |
| 13 | Интерфейс емкости | Нажать «Назад»   | Главное меню      | Возврат в меню выбора моделей  |

Основные функциональные проверки включают:

- корректное функционирование вращения трехмерной модели (запуск вращения в обе стороны, мгновенная смена направления, остановка при повторном нажатии);
- загрузку и корректное отображение трехмерной модели оборудования;
- корректную работу системы проверки знаний по заданному параметру для каждой модели с отображением соответствующих сообщений;
- стабильность работы при переключении между различными моделями оборудования;
- корректное отображение подписанных элементов на трехмерной модели;
- корректное отображение статичной SCADA-схемы для каждой модели оборудования.

Прохождение 9 пункта сценария представлено на рисунке 2.

## Практика

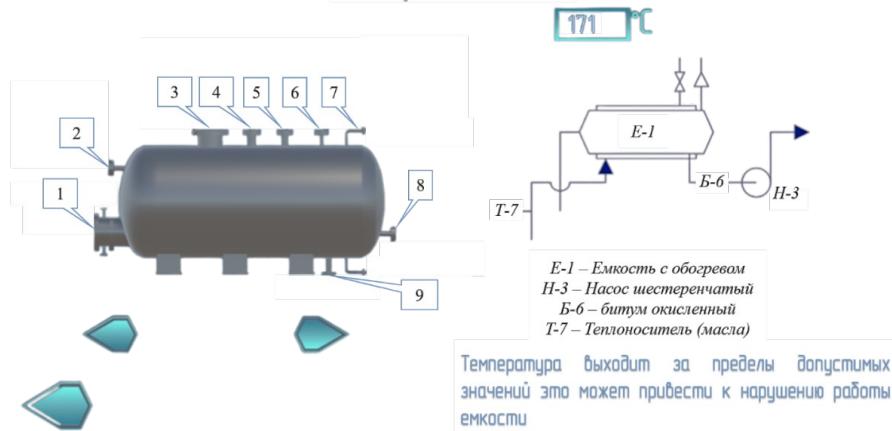


Рисунок 2 - Прохождение 9-ого пункта сценария:

1 - камера распределительная; 2 - штуцер перелива; 3 - люк-лаз; 4 – штуцер отбора проб; 5 - штуцер ввода битума; 6 - выход воздуха (пара); 7 - штуцер для уровнемера; 8 - штуцер вывода битума; 9 – штуцер спуска остатка

DOI: <https://doi.org/10.60797/IRJ.2026.164.17.3>

Создание подобного сценария позволит не ограничиваться внешней проверкой. Например, в Unity Test Framework можно внедрить автотесты, отслеживающие корректность работы скриптов взаимодействия. Такой подход экономит время и повышает стабильность системы после обновлений.

Для сложных трехмерных моделей с высокой детализацией важно проводить визуальное тестирование вручную, так как автоматические инструменты пока не способны полноценно оценить качество рендеринга, отражений и реалистичность анимации.

Кроме того, полезно создавать отдельные сценарии для проверки стабильности трехмерного рендерера, особенно если используется физически корректное освещение. Иногда достаточно небольшой ошибки в скрипте или конфликтов между материалами, чтобы нарушить отображение всей модели.

### Обработка результатов тестирования и классификация дефектов

После выполнения функционального тестирования по сценарию составляется акт выявленных несоответствий, который позволяет внести корректировку в программное приложение [10]. Для обучающих программных приложений типичными проблемами являются несоответствие логики обучения и поведения модели, неправильная обработка пользовательских событий или сбои в синхронизации звука и графики.

На примере рассмотрим классификацию дефектов, представленных в таблице 2.

Таблица 2 - Классификация дефектов

DOI: <https://doi.org/10.60797/IRJ.2026.164.17.4>

| Уровень критичности    | Пример ошибки  | Влияние на обучение   | Приоритет устранения       |
|------------------------|--|---|----------------------------|
| Критические дефекты    | Приложение не запускается, модель не отображается, обучение невозможно | Обучение невозможно, система не выполняет основную функцию          | Немедленный (1 уровень)    |
| Средние дефекты        | Отдельные функции работают некорректно                                 | Частично нарушается логика обучения или взаимодействие пользователя | Высокий (2 уровень)        |
| Незначительные дефекты | Визуальные неточности, не влияющие на учебный процесс                  | Не влияют на процесс обучения, но снижают качество восприятия       | Средний/низкий (3 уровень) |

Использование систем управления дефектами, таких как JIRA, TestRail или Redmine позволяет структурировать процесс тестирования. Каждое найденное отклонение фиксируется с указанием версии, модуля и условий воспроизведения.

На практике также важно выстраивать обратную связь между тестировщиками и разработчиками. Например, после каждой итерации тестирования в Unity можно автоматически генерировать отчет о дефектах в формате CSV, где фиксируются: время возникновения ошибки, описание, версия сборки, шаги воспроизведения. Это позволяет быстро отследить повторяющиеся ошибки и исключить их в будущем.



## Заключение

Проведенное исследование подтвердило важность системного функционального тестирования для обеспечения надежности программных приложений обучающих трехмерных моделей технологического оборудования.

Полученные результаты и их новизна:

Разработан детальный сценарий функционального тестирования для блока модели «Емкость», который охватывает ключевые пользовательские операции: вращение модели, ввод и валидацию технологических параметров, корректность отображения элементов и SCADA-схем. Сценарий построен с применением техник анализа граничных значений и тестирования переходов состояний, что повышает вероятность выявления скрытых дефектов бизнес-логики.

Предложена классификация дефектов, специфичная для обучающих приложений, с разделением по уровню критичности и влиянию на учебный процесс. Это позволяет отслеживать исправления и управлять качеством продукта.

Методика носит типовой характер и может быть адаптирована для тестирования других модулей и аналогичных программных комплексов в нефтегазовой и других отраслях.

Оригинальность работы заключается в адаптации классического метода «черного ящика» к задачам тестирования интерактивных трехмерных моделей технологического оборудования, что ранее в литературе освещалось недостаточно. Разработанный сценарий и принципы методики ориентированы именно на конечного пользователя, что гарантирует проверку функциональности в условиях, максимально приближенных к реальной эксплуатации.

Таким образом, реализация предложенной методики позволяет гарантировать техническую корректность и удобство использования обучающего приложения, что напрямую способствует повышению качества подготовки специалистов для нефтегазовых отраслей.

## Конфликт интересов

Не указан.

### Рецензия

Рудой Е.М., ООО «ГК «Иннотех», Москва Российская Федерация

DOI: <https://doi.org/10.60797/IRJ.2026.164.17.5>

## Conflict of Interest

None declared.

### Review

Rudoi E.M., Innotech Group LLC, Moscow Russian Federation

DOI: <https://doi.org/10.60797/IRJ.2026.164.17.5>

## Список литературы / References

- ГОСТ Р ИСО/МЭК 20000-1-2021. Требования к системе менеджмента сервисов. — Введ. 2021-12-07. — Москва: Российский институт стандартизации, 2021. — 32 с.
- 2025 State of Digital Quality Report From Applause Reveals Sharp Increase in AI-Powered Functional Testing, but Human Involvement Remains Critical. — 2025. — URL: [https://www.applause.com/press-release/2025-state-of-digital-quality-functional-testing/?utm\\_source=chatgpt.com](https://www.applause.com/press-release/2025-state-of-digital-quality-functional-testing/?utm_source=chatgpt.com) (accessed: 14.11.25)
- ГОСТ Р 56939-2024. Защита информации. Разработка безопасного программного обеспечения. Общие требования. — Введ. 2024-12-20. — Москва: Стандартинформ, 2024. — 26 С.
- Гунина Е.В. Метод визуализации результатов UX-тестирования информационного ресурса / Е.В. Гунина, С.В. Иванова // Актуальные проблемы инфотелекоммуникаций в науке и образовании. — Санкт-Петербург: СПбГУТ им. проф. М.А. Бонч-Бруевича, 2021.
- Гусеница Я.Н. Методика оценивания полноты результатов испытаний / Я.Н. Гусеница. // Состояние и перспективы развития современной науки по направлению «АСУ, информационно-телекоммуникационные системы»; — Анапа: ЭРА, 2021.
- Уймин А.Г. Разработка методики тестирования системы безопасности автоматизированных систем управления технологическими процессами на основе корпоративного стандарта / А.Г. Уймин // Автоматизация и информатизация ТЭК. — 2024. — № 5.
- Хабрат В.В. Методика испытаний адресно-аналоговых систем пожарной сигнализации / В.В. Хабрат // Вестник науки. — 2025. — № 7.
- Черепнин М.А. Комплексное сравнение методов при выборе подходящего метода процесса тестирования программного обеспечения / М.А. Черепнин // Научно-исследовательский центр «Technical Innovations». — 2023. — № 18.
- Кустов Д.А. Анализ тестирования программного обеспечения методом белого ящика и методом черного ящика / Д.А. Кустов // Научный аспект. — 2024. — № 5.
- Кузьменков К.А. Анализ методов тестирования программного обеспечения: от функциональных проверок до исследовательского тестирования / К.А. Кузьменков. // Цифровая экономика глазами студентов; — Казань: Сагиев А.Р., 2025.

## Список литературы на английском языке / References in English

- GOST R ISO/IEC 20000-1-2021. Trebovaniya k sisteme menedzhmenta servisov [Requirements for the service management system]. — Introduced 2021-12-07. — Moscow: Russian Institute of Standardization, 2021. — 32 p. [in Russian]

2. 2025 State of Digital Quality Report From Applause Reveals Sharp Increase in AI-Powered Functional Testing, but Human Involvement Remains Critical. — 2025. — URL: [https://www.applause.com/press-release/2025-state-of-digital-quality-functional-testing/?utm\\_source=chatgpt.com](https://www.applause.com/press-release/2025-state-of-digital-quality-functional-testing/?utm_source=chatgpt.com) (accessed: 14.11.25)
3. GOST R 56939-2024. Zashchita informacii. Razrabotka bezopasnogo programmnogo obespecheniya. Obshchie trebovaniya [GOST R 56939-2024. Information protection. Development of secure software. General requirements]. — Introduced 2024-12-20. — Moscow: Standartinform, 2024. — 26 P. [in Russian]
4. Gunina Ye.V. Metod vizualizatsii rezul'tatov UX-testirovaniya informatsionnogo resursa [Method of visualization of UX testing results of an information resource] / Ye.V. Gunina, S.V. Ivanova // Current Problems of Infotelecommunications in Science and Education. — Saint Petersburg: M.A. Bonch-Bruevich St. Petersburg State University, 2021. [in Russian]
5. Gusenica Ya.N. Metodika ocenivaniya polnoty' rezul'tatov ispy'tanij [Methodology for assessing the completeness of test results] / Ya.N. Gusenica. // State and Prospects of Development of Modern Science in the Field of Automated Control Systems, Information and Telecommunication Systems; — Anapa: E'RA, 2021. [in Russian]
6. Ujmin A.G. Razrabotka metodiki testirovaniya sistemy' bezopasnosti avtomatizirovanny'x sistem upravleniya texnologicheskimi processami na osnove korporativnogo standarta [Development of a methodology for testing the security system of automated process control systems based on a corporate standard] / A.G. Ujmin // Automation and Informatization of the Fuel and Energy Complex. — 2024. — № 5. [in Russian]
7. Xabrat V.V. Metodika ispy'tanij adresno-analogovy'x sistem pozharnoj signalizacii [Methodology for testing addressable analog fire alarm systems] / V.V. Xabrat // Science Bulletin. — 2025. — № 7. [in Russian]
8. Cherepnin M.A. Kompleksnoe sravnenie metodov pri vy'bore podxodyashhego metoda processa testirovaniya programmnogo obespecheniya [Comprehensive comparison of methods when choosing the appropriate method for the software testing process] / M.A. Cherepnin // Scientific Research Center "Technical Innovations". — 2023. — № 18. [in Russian]
9. Kustov D.A. Analiz testirovaniya programmnogo obespecheniya metodom belogo yashhika i metodom chernogo yashhika [Analysis of software testing by the white box method and the black box method] / D.A. Kustov // Scientific Aspect. — 2024. — № 5. [in Russian]
10. Kuz'menkov K.A. Analiz metodov testirovaniya programmnogo obespecheniya: ot funkcional'ny'x proverok do issledovatel'skogo testirovaniya [Analysis of software testing methods: from functional checks to research testing] / K.A. Kuz'menkov. // Digital Economy Through the Eyes of Students; — Kazan': Sagiev A.R., 2025. [in Russian]