

ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.415.538

ПРИМЕНЕНИЕ SWOT-АНАЛИЗА В ТЕСТИРОВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Галимова Екатерина Юрьевна, кандидат технических наук, доцент кафедры математического обеспечения и применения ЭВМ, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, Россия
e-mail: galim81@mail.ru

Галимова Анастасия Тимуровна, студент, направление подготовки 09.03.01 Информатика и вычислительная техника, Санкт-Петербургский государственный морской технический университет, Санкт-Петербург, Россия
e-mail: galimova463@gmail.com

Аннотация. В статье рассматривается применение SWOT-анализа как инструмента стратегического планирования в сфере тестирования программного обеспечения. Проведён анализ, в ходе которого оценены сильные стороны (опыт команды, автоматизация, быстрая обратная связь) рассматриваемого процесса и его слабые места (недостаточное покрытие тестами, устаревшие инструменты, слабая документация). Также определены возможности для развития (новые инструменты, обучение, интеграция с CI/CD) и проанализированы внешние угрозы (рост требований, кадровые риски, усиление конкуренции). Предложены направления для оптимизации процессов тестирования и повышения качества программных продуктов.

Ключевые слова: SWOT-анализ, тестирование программного обеспечения, автоматизация тестирования, управление качеством, инструменты тестирования, стратегическое планирование.

Для цитирования: Галимова Е. Ю., Галимова А. Т. Применение SWOT-анализа в тестировании программного обеспечения // Шаг в науку. – 2026. – № 3. – С. 5–8.

APPLICATION OF SWOT ANALYSIS IN SOFTWARE TESTING

Galimova Ekaterina Yuryevna, Candidate of Technical Sciences, Associate Professor of the Department of Mathematical Software and Computer Applications, Saint Petersburg Electrotechnical University, Saint Petersburg, Russia
e-mail: galim81@mail.ru

Galimova Anastasia Timurovna, student, training program 09.03.01 Computer Science and Engineering, Saint Petersburg State Marine Technical University, Saint Petersburg, Russia
e-mail: galimova463@gmail.com

Abstract. The article discusses the application of SWOT analysis as a strategic planning tool in the field of software testing. The analysis evaluates the strengths (team experience, automation, and quick feedback) and weaknesses (insufficient test coverage, outdated tools, and poor documentation) of the testing process. It also identifies opportunities for development (new tools, training, and integration with CI/CD) and analyzes external threats (increasing requirements, personnel risks, and increased competition). The article suggests ways to optimize testing processes and improve the quality of software products.

Key words: SWOT analysis, software testing, test automation, quality management, testing tools, strategic planning.

Cite as: Galimova, E. Yu., Galimova, A. T. (2026) [Application of SWOT analysis in software testing]. *Shag v nauku* [Step into science]. Vol. 3, pp. 5–8.

Одним из популярных инструментов стратегического планирования на сегодняшний день является SWOT-анализ [1]. Рассмотрим возможности его применения в сфере тестирования программного обеспечения. Его суть заключается в структурированной оценке четырёх ключевых аспектов: сильных и слабых сторон продукта или процесса тестирования, а также возможностей и угроз со стороны внешней среды.

Применим SWOT-анализ в тестировании программного обеспечения. На первом шаге проведем оценку внутренних процессов тестирования. Выделим сильные стороны. *Опытная команда* – это специалисты с глубокими знаниями в области тестирования, которые умеют быстро находить и анализировать дефекты, предлагать улучшения и оптимизировать процессы. Опыт позволяет минимизировать риски пропуска критических ошибок и эффективно работать в условиях сжатых сроков. *Автоматизация* ускоряет проверку функционала, снижает влияние человеческого фактора, повышает стабильность и повторяемость проверок. Она особенно эффективна для регрессионного тестирования и позволяет быстрее выпускать релизы. *Быстрая обратная связь* – это оперативное информирование разработчиков о найденных ошибках и результатах тестирования. Наличие такой связи способствует быстрому исправлению дефектов, улучшает взаимодействие между командами и повышает качество продукта.

Далее выделим слабые стороны. *Недостаток тестов* или ограниченное покрытие тестами приводит к тому, что часть функционала остаётся непроверенной, что увеличивает вероятность пропуска дефектов [4]. Возникновение такой проблемы может быть связано с нехваткой времени, ресурсов или недостаточным вниманием к тестированию на ранних этапах. *Устаревшие инструменты* – это программы, фреймворки или методики, которые больше не поддерживаются, редко используются в современной практике или заменены более эффективными и удобными аналогами. Такие инструменты могут не получать обновлений, не поддерживать новые технологии или уступать по функционалу и производительности современным решениям. Использование старых версий инструментов тестирования снижает эффективность работы, ограничивает возможности автоматизации, усложняет интеграцию с новыми системами и может привести к техническим проблемам. *Слабая документация* не обеспечивает достаточной поддержки для тестировщиков, разработчиков и других участников проекта. Основные виды документации, используемой в процессе тестирования программного обеспечения, указаны в таблице 1. Отсутствие или низкое качество тестовой документации затрудняет понимание требований, усложняет передачу знаний между сотрудниками, увеличивает время на обучение новых специалистов и снижает прозрачность процессов.

Таблица 1. Виды тестовой документации

Вид документации	Краткое описание
План тестирования	Описывает цели, стратегию, ресурсы, сроки, риски и объем тестирования
Тест-кейсы	Пошаговые сценарии проверки отдельных функций с ожидаемым результатом
Инструкции по окружению	Описание, как развернуть тестовую среду
Наборы тестов	Группы связанных тестов по теме или функции
Чек-листы	Простые списки проверок, где отмечается выполнение каждого пункта
Баг-репорты	Документы о найденных ошибках с описанием, шагами воспроизведения и окружением
Отчеты о тестировании	Итоговые документы с результатами: сколько тестов пройдено, сколько ошибок найдено и т. д.
Матрица трассируемости	Таблица соответствия между требованиями и тестами, показывающая полноту покрытия
Планы по уровням тестирования	Документы для модульного, интеграционного, системного и приемочного тестирования

Источник: составлено авторами

На следующем шаге рассмотрим возможности для улучшения процесса тестирования. *Новые инструменты* тестирования, например, платформы для тестирования чат-ботов, инструменты для контрактного тестирования (для интеграции микросервисов), применение ИИ-агентов для автоматизации тестирования, позволяют вывести процесс тестирования на новый уровень [3]. Внедрение современных средств тестирования и управления дефектами дает возможность повысить качество и скорость работы, улучшить аналитику и отчетность, а также автоматизировать рутинные задачи, что представлено

в таблице 2. *Повышение квалификации* сотрудников помогает актуализировать и структурировать знания и навыки [6]. Полезны тренинги, курсы и обмен опытом, способствующие росту профессионализма, внедрению лучших практик и инноваций в процесс тестирования. *Интеграция с CI/CD* является очень перспективной практикой [2]. Встраивание тестирования в процессы непрерывной интеграции и доставки (CI/CD) обеспечивает автоматический запуск тестов при каждом изменении кода, что позволяет выявлять ошибки на ранних стадиях и ускоряет выпуск новых версий продукта [5; 7].

Таблица 2. Новые инструменты/подходы для тестирования

Направление	Примеры инструментов/подходов	Ключевые преимущества
ИИ-агенты	ИИ-приоритизация, автообновление тестов testRigor, Selenium	Сокращение времени регрессионного тестирования
Тестирование ИИ/LLM	Платформы для проверки чат-ботов Botium, Postman, Dialogflow	Безопасность, этичность
API/контрактное тестирование	Инструменты Postman, Karakum, Spring Cloud Contact	Стабильность интеграций
No-code решения	Инструменты: Workbeaver AI, Zapier, Tally	Не требуется написание кода
Облачные платформы тестирования производительности	Платформы BlazeMeter, LoadNinja	Эмуляция высоких нагрузок из разных точек мира без покупки серверов

Источник: составлено авторами

Перейдем к рассмотрению существующих угроз. *Рост требований* может стать серьезным риском для проекта, что представлено в таблице 3. Увеличение объема и сложности функционала, а также повышение ожиданий пользователей требуют более тщательного и масштабного тестирования, что может привести к нехватке ресурсов и времени. В результате возможен срыв сроков выхода продукта на рынок, что негативно сказывается на конкурентоспособности компании. *Кадровые риски* – это угрозы, связанные с человеческим фактором. Уход ключевых специали-

стов, нехватка квалифицированных тестировщиков или высокая текучесть кадров могут негативно сказаться на качестве и стабильности процессов тестирования. *Конкуренция* в тестировании программного обеспечения может быть как двигателем прогресса, так и источником ряда рисков и негативных последствий для компаний и продуктов. Появление новых игроков на рынке, усиление позиций существующих конкурентов требуют постоянного повышения качества продукта и скорости его вывода на рынок, что увеличивает нагрузку на команду тестирования.

Таблица 3. Последствия роста числа требований в проекте

Риск	Описание
Увеличение объема тестирования	Больше тестов – больше расход времени и ресурсов
Задержка релиза	Срыв сроков выхода продукта
Снижение качества	Рост числа пропущенных дефектов
Рост затрат	Дорогостоящие доработки и исправления
Отказ от тестирования	Вероятен выпуск продукта с ошибками
Сложности коммуникации	Недопонимание между участниками проекта

Источник: составлено авторами

Каждому фактору рекомендуется присвоить баллы по степени влияния, чтобы определить приоритеты для дальнейших действий. Далее составляется план, какие слабые стороны можно устранить, как использовать возможности и минимизировать угрозы. Пример плана действий:

- усилить автоматизацию;
- обучить команду новым инструментам;
- пересмотреть процессы коммуникации с раз-

работчиками;

- заранее подготовиться к возможным изменениям требований.

SWOT-анализ в тестировании программного обеспечения стал не только способом выявить текущие проблемы, но и инструментом для стратегического развития QA-процессов, повышения качества продукта и снижения рисков на всех этапах жизненного цикла разработки.

Литература

1. Булов А. А., Кислова Т. А. Методы и модели стратегического анализа: Учебное пособие. – СПб.: Научные технологии. – 2025. – 202 с. – EDN: YHTFON.
2. Галимова Е. Ю. Особенности проведения тестирования программного обеспечения в рамках гибкой методологии разработки // Актуальные вопросы современной науки и образования: материалы Всероссийской научно-практической конференции, посвященной 25-летию Нефтекамского филиала УУНиТ, Нефтекамск, 24 октября 2025 года. – Уфа: Уфимский университет науки и технологий. – 2025. – С. 62–63. – EDN: HGCWRX.
3. Галимова Е. Ю. Современные векторы развития процесса тестирования программного обеспечения // Инновации в информационных технологиях, машиностроении и автотранспорте: Сборник материалов VIII Международной научно-практической конференции, Кемерово, 21–23 ноября 2024 года. – Кемерово: Кузбасский государственный технический университет им. Т.Ф. Горбачева. – 2024. – С. 149–152. – EDN: KYUBFC.
4. Майерс Г., Баджет Т., Сандлер К. Искусство тестирования программ, 3-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс». – 2016. – 272 с.
5. Рубин Кеннет С. Основы Scrum: практическое руководство по гибкой разработке ПО: пер с англ. – СПб.: ООО «Диалектика». – 2020. – 544 с.
6. Тюлькин И. В. QA: тестирование, автоматизация и процессы на экспертном уровне. – М.: ДМК Пресс. – 2025. – 184 с.
7. Хамбл Дж., Фарли Д. Непрерывное развертывание ПО. Автоматизация процессов сборки, тестирования и внедрения новых версий программ: Пер. с англ. – М.: ООО «И.Д. Вильямс». – 2017. – 432 с.

Статья поступила в редакцию: 08.04.2026; принята в печать: 05.08.2026.

Авторы прочитали и одобрили окончательный вариант рукописи.