

ROI АВТОМАТИЗАЦИИ: ЭКОНОМИКО-МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЦЕЛЕСООБРАЗНОСТИ

Кондратенко Е.В., Шиш П.А.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Научный руководитель: Коваленко И.В. – магистр экономических наук, ст. преподаватель кафедры ПИКС

Аннотация. Рассматривается задача оценки целесообразности внедрения автоматизированного тестирования в учебное веб-приложение для коммерческой библиотеки. Предложена экономико-математическая модель расчета точки окупаемости (*break-even point*) и ROI автоматизации, учитывающая временные и инвестиционные затраты на ручное тестирование и разработку автотестов с настройкой инфраструктуры, а также экономический эффект от сокращения времени регрессионной проверки и частичного снижения стоимости дефектов за счет более раннего обнаружения.

Ключевые слова: ручное тестирование, автоматизация тестирования, *Java*, *Spring Boot*, *JUnit*, интеграционное тестирование, *Jenkins*, точка окупаемости, ROI

Введение. В качестве объекта исследования рассматривается курсовой проект «Аналитическое приложение для автоматизации учета новых поступлений и мониторинга выдачи-возврата книг читателями в библиотеке» с клиент-серверной архитектурой в виде REST API-сервиса. Контейнерный уровень проекта содержит *frontend*-часть (*HTML5*, *CSS*, *TypeScript*, *Bootstrap*, *Angular*) и *backeng*-часть (*Java*, *Spring Boot*, *PostgreSQL*).

Для тестирования: *Windows 11 (x64)*, *JDK 17*, *Apache Maven*, *Spring Boot Starter Test*, *Maven Surefire Plugin 3.2.5* (запуск модульных тестов), *JaCoCo Maven Plugin 0.8.9* (анализ покрытия кода тестами), *Maven Checkstyle Plugin 3.3.1* (проверка соответствия кода стандартам), *Jenkins* (автоматический запуск сборки, тестов и проверок качества кода), *Docker* и *Docker Compose* (развертывание тестовой среды и сервисов базы данных).

В данной статье авторы представляют результаты эксперимента в формате «до» и «после» внедрения автотестов и настройки окружения. Измерения фиксировались на одинаковом наборе регрессионных сценариев и одинаковом аппаратном (CI-окружении), чтобы сравнение отражало именно вклад автоматизации.

Основная часть. Для оценки тестового покрытия кода серверной части приложения использовалась *JaCoCo* – специализированная библиотека покрытия кода, разработанная командой *EclEmma* [1]. Для большинства компонентов проекта показатель покрытия находится в диапазоне 85-100 %, а общее покрытие инструкций по системе составило 91 %.

Для оценки экономической эффективности автоматизации тестирования используется показатель ROI (*Return on Investment*) – отношение полученной выгоды к затратам на внедрение [2]. Показатель рассчитывается по формуле 1:

$$ROI = \frac{B-C}{C} \cdot 100\%, \quad (1)$$

где *B* – суммарные выгоды (*benefits*);

C – суммарные затраты (*cost*).

В рамках проекта рассматривалась автоматизация регрессионного тестирования серверной части приложения. В качестве выгоды учитывалась экономия времени выполнения тестов, а затратами – трудозатраты на разработку тестов и их поддержку.

Пусть задан период анализа – год, и определены:

– *r* – стоимость одного человеко-часа (руб./ч) для работ по тестированию/разработке (включая налоги и накладные);

Направление «Электронные системы и технологии»

- N – число регрессионных прогонов за период;
 - t_{man} – средняя длительность одного ручного прогона регрессии (включая подготовку окружения и выполнение);
 - t_{auto} – длительность одного автоматизированного прогона (включая запуск и минимальный анализ результатов);
 - t_{maint} – средние трудозатраты поддержки автотестов на один цикл (например, на релиз/спринт);
 - T_{init} – трудозатраты разработки автотестов и инфраструктуры (часы).
- Экономический эффект от сокращения «ручного» труда вычисляется по формуле 2:

$$B = N \cdot \Delta t \cdot r. \quad (2)$$

Тогда суммарные затраты вычисляются по формуле 3:

$$C = T_{init} \cdot r + N \cdot t_{maint} \cdot r. \quad (3)$$

В ходе работы с проектом получены следующие значения:

- ручной прогон: подготовка окружения и данных $t_{env} = 0,42$ ч и прохождение сценариев $t_{exec} = 2,00$ ч, $t_{man} = 2,42$ ч;
 - автоматизированный прогон: старт контейнера БД и запуск интеграционных, API-тестов $0,25$ ч и анализ отчета, логов $0,10$ ч, $t_{auto} = 0,35$ ч;
 - поддержка автотестов на цикл $t_{maint} = 0,40$ ч (корректировка тестов при изменении API, валидаций, ролей);
 - стоимость часа $r = 45$ руб./ч;
 - одноразовые трудозатраты на внедрение (T_{init}) приняты 60 ч (настройка тестовой архитектуры, написание набора *unit* и интеграционных тестов, интеграция в CI).
- Экономия времени на каждом цикле вычислена по формуле 4:

$$\Delta t = t_{man} - t_{auto} = 2,42 - 0,35 = 2,07 \text{ ч.} \quad (4)$$

Накопленная чистая выгода на один цикл рассчитана по формуле 5:

$$g = (\Delta t - t_{maint}) \cdot r = (2,07 - 0,40) \cdot 45 = 75,15 \text{ руб.} \quad (5)$$

Начальные затраты на внедрение автоматизации вычислена по формуле 6:

$$T_{init} \cdot r = 60 \cdot 45 = 2700 \text{ руб.} \quad (6)$$

Точка окупаемости по числу циклов регрессии вычислена по формуле 7:

$$N^* = \frac{T_{init} \cdot r + C_{tool}}{g} = \frac{2700}{75,15} \approx 36 \text{ циклов.} \quad (7)$$

Интерпретация рассчитанного показателя критически зависит от частоты регрессий, поскольку фактор «как часто тест нужно запускать» выделяется как ключевой критерий целесообразности автоматизации: чем больше релизов (циклов), тем выше выгода от автоматизации регрессии [3].

Рассмотрим два режима эксплуатации:

- 1 При редких прогонах (1 регрессия в месяц $N = 12$).
- 2 При частых прогонах (1 регрессия в неделю, $N = 52$).

Используя формулы выше получены значения для представленных ситуаций, включая *ROI* (таблица 1):

Таблица 1 – Сравнительный анализ показателей *ROI* автоматизации тестирования для различных режимов регрессионного тестирования

Показатель	Режим А: $N = 12$	Режим В: $N = 52$
Выгода по времени B	1117,8 руб.	4843,8 руб.
Затраты на поддержку $N \cdot t_{maint} \cdot r$	216 руб.	936 руб.
Общие затраты C	2916 руб.	3636 руб.
<i>ROI</i> по времени	-61,7 %	+33,2 %

Исходя из полученного значения при редких прогонах регрессии автоматизация может не окупиться на годовом горизонте; при частых прогонах – становится экономически целесообразной (положительный *ROI*, достижение точки окупаемости приблизительно на 36-м цикле).

Заключение. Проведенные расчеты показателей *ROI* и точки безубыточности показали, что автоматизация тестирования требует больших начальных трудозатрат по сравнению с ручным тестированием. Однако при дальнейшем использовании автоматизированные тесты позволяют заметно снизить затраты на выполнение повторных регрессионных проверок. При редком запуске регрессии автоматизация может не окупиться в пределах годового периода. Экономическая эффективность автоматизации напрямую зависит от того, как часто выполняются регрессионные проверки и насколько активно развивается программный продукт. Предложенная модель расчета может применяться при планировании тестирования в аналогичных проектах на базе *Spring*.

Список литературы

1. Отчеты о покрытии кода JUnit тестами с помощью Maven плагина JaCoCo [Электронный ресурс]. Режим доступа: <https://habr.com/ru/articles/591553/>. Дата доступа: 03.03.2026.
2. What Is Return on Investment (ROI) and How to Calculate It [Электронный ресурс]. Режим доступа: <https://www.investopedia.com/terms/r/returnoninvestment.asp>. Дата доступа: 03.03.2026.
3. Automation Testing [Электронный ресурс]. Режим доступа: <https://www.guru99.com/automation-testing.html>. Дата доступа: 05.03.2026.

UDC 004.415.538

ROI OF AUTOMATION: ECONOMIC AND MATHEMATICAL MODEL OF FEASIBILITY

Kondratenko E.V., Shish P.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Kovalenko I.V. – head of the Department of Innovative Developments in Education, Master of Sci. (M. of Sci.), Senior Lecturer at the Department of ICSD

Annotation. The task of assessing the feasibility of implementing automated testing in an academic web application for a commercial library is considered. An economic and mathematical model for calculating the break-even point and ROI of automation is proposed, taking into account the time and investment costs of manual testing and development of autotests with infrastructure configuration, as well as the economic effect of reducing the time of regression testing and partially reducing the cost of defects due to earlier detection.

Keywords: manual testing, test automation, Java, Spring Boot, JUnit, integration testing, Jenkins, break-even point, ROI