

УДК 004.4

DOI: [10.26102/2310-6018/2025.51.4.011](https://doi.org/10.26102/2310-6018/2025.51.4.011)

Система оценки продолжительности жизненного цикла разработки программного обеспечения на основе интеллектуальной обработки информации

Д.Э. Храмов✉

*Саратовский государственный технический университет имени Гагарина Ю.А.,
Саратов, Российская Федерация*

Резюме. В статье представлена система оценки продолжительности жизненного цикла разработки программного обеспечения, основанная на применении технологий искусственного интеллекта. Проведен анализ существующих подходов к оценке трудозатрат и сроков разработки, на основе которого обоснован выбор нейросетевых технологий как наиболее перспективного направления для решения задач прогнозирования в условиях неопределенности. Выделены и классифицированы основные группы факторов, оказывающих влияние на длительность процесса разработки: технические, организационные, командные, исторические, ресурсные, внешние. На основе классов факторов определены множества входных параметров, используемых для обучения нейронных сетей, а также их гиперпараметры. Приведены архитектурные характеристики нейронных сетей, включая количество слоев, типы функций активации, методы оптимизации и параметры регуляризации, обученных в ходе экспериментов. Разработан алгоритм оценки сроков, реализованный в виде программной системы, обеспечивающей автоматизированное прогнозирование продолжительности разработки проектов на основе анализа исторических данных и текущих характеристик проекта. Приведен пример проведения оценки сроков разработки с помощью разработанной системы и проведено сравнение полученных результатов с экспертной оценкой. Предложенная система сокращает длительность анализа и повышает точность оценки по сравнению с традиционными методами.

Ключевые слова: нейронная сеть, жизненный цикл разработки программного обеспечения, прогнозирование сроков, программная система, программная инженерия.

Для цитирования: Храмов Д.Э. Система оценки продолжительности жизненного цикла разработки программного обеспечения на основе интеллектуальной обработки информации. *Моделирование, оптимизация и информационные технологии.* 2025;13(4). URL: <https://moitvvt.ru/ru/journal/pdf?id=2045> DOI: 10.26102/2310-6018/2025.51.4.011

Software development life cycle duration assessment system based on intelligent information processing

D.E. Khramov✉

Yuri Gagarin State Technical University of Saratov, Saratov, the Russian Federation

Abstract. The article presents a system for assessing the durability of the software development life cycle based on the use of artificial intelligence technologies. An analysis of existing approaches to the science of labor costs and development times is presented, based on which the choice of neural network technologies is substantiated as the most promising direction for solving forecasting problems under uncertainty. The main groups of factors influencing the duration of the development process are identified and classified: technical, organizational, team, historical, resource, external. Based on the classes of factors, constant distribution of input parameters, application for training neural networks, as well as their hyperparameters. The architectural characteristics of neural networks, the number of layers, types of activation functions, optimization methods and control parameters studied in the experiments are given. An algorithm for assessing the timing has been developed, implemented as a software system

that provides operational forecasting of the durability of project development based on the analysis of historical data and current project analytics. An example of assessing the development times using the developed system is given and the results are compared with an expert assessment. The proposed system for analyzing the duration of the reduction and increasing the accuracy of the estimate in comparison with the reduction methods.

Keywords: neural network, software development life cycle, time estimating, software system, software engineering.

For citation: Khramov D.E. Software development life cycle duration assessment system based on intelligent information processing. *Modeling, Optimization and Information Technology*. 2025;13(4). (In Russ.). URL: <https://moitvvt.ru/ru/journal/pdf?id=2045> DOI: 10.26102/2310-6018/2025.51.4.011

Введение

Успешная разработка программного обеспечения (ПО) определяется рядом ключевых критериев, включая соблюдение сроков, соответствие бюджету и реализацию заявленного функционала. Оценка сроков разработки ПО является неотъемлемой частью успешного проекта. В случае, если прогнозируемый срок разработки оказался меньше реализуемого, то компаниям необходимо вкладывать дополнительные средства на продолжение разработки или выпускать продукт, который способен реализовать только часть функционала, что, в свою очередь, влияет на репутацию компании.

В настоящее время существуют различные подходы к оценке сроков разработки ПО, которые можно классифицировать следующим образом:

- алгоритмы, основанные на математическом расчете;
- алгоритмы с экспертами;
- алгоритмы с историческими данными;
- графический метод;
- модифицированные алгоритмы [1].

Среди указанных выше классов наибольший интерес представляют алгоритмы с экспертами. Данный класс основан на оценке людей-экспертов, которые коллективно вычисляют срок, необходимый на реализацию ПО.

Данный подход имеет несколько вариантов реализации с точки зрения экспертов: в качестве эксперта выступает один сотрудник компании; в качестве экспертов выступают несколько сотрудников компании; в качестве экспертов выступает сторонняя компания, которая занимается экспертной оценкой. Кроме этого, существуют различные методы проведения оценки с экспертом: метод покер планирования, который используется для Agile-методологий, а также УСР, метод Дельфи.

Стоит отметить, что у алгоритмов с экспертами есть важный недостаток – срок оценки разработки ПО. В определенных случаях на проведение оценки сторонними компаниями необходимо от нескольких дней до нескольких недель общения с менеджерами и командой разработки. Кроме этого, необходимо отметить, что в Российской Федерации в компаниях в роли эксперта находятся сами сотрудники компании.

Другими недостатками алгоритма являются его субъективность и зависимость от квалификации эксперта. Если сотрудник компании из команды А будет проводить оценку сроков разработки командой Б, то оценка экспертом будет опираться на его собственный опыт работы с командой А.

Следовательно, можно сделать вывод, что данный подход характеризуется низкой эффективностью с точки зрения затрачиваемого времени.

Материалы и методы

Для вычисления сроков разработки ПО выбранным алгоритмом возможно использовать следующие классы факторов, представленные в Таблице 1.

Таблица 1 – Классы факторов, влияющие на анализ продолжительности жизненного цикла ПО
Table 1 – Classes of factors influencing the analysis of software life cycle duration

| Класс факторов | Примеры |
|-----------------|--|
| Технические | Function Points (FP), масштаб проекта, тип архитектуры |
| Организационные | Методология, частота изменений |
| Командные | Размер команды, опыт, вовлечённость |
| Исторические | Данные по предыдущим проектам |
| Ресурсные | Бюджет, доступность инструментов |
| Внешние | Требования, риски, неопределённость |

В настоящее время в ряде российских IT-компаний используют наборы данных с меньшим количеством факторов.

Для решения данной проблемы была разработана модель представления данных для вычисления анализа продолжительности жизненного цикла (ЖЦ) ПО. Формальная модель задачи анализа продолжительности ЖЦ ПО представлена по формуле:

$$T_{\text{прогноз}} = f(x_1, x_2, \dots, x_n), \quad (1)$$

где x_1, x_2, \dots, x_n – факторы проекта, а $f(\cdot)$ – функция прогнозирования.

Следующие множества описывают соответствующие классы факторов, которые используются для анализа продолжительности ЖЦ ПО в рассматриваемом алгоритме:

$X_{\text{техн}} = \{FP, \text{масштаб проекта, сложность проекта}\}$, где $FP \geq 1$, масштаб проекта $\in [0,5]$, сложность проекта $\in [0,5]$.

$X_{\text{ком}} = \{\text{опыт команды, скорость работы}\}$, где опыт команды $\in [0,5]$, скорость работы $\geq 0,1$.

$X_{\text{ист}} = \{T_{\text{прошлые}}, FP_{\text{прошлые}}\}$, где $T_{\text{прошлые}} \geq 1$, $FP_{\text{прошлые}} \geq 1$.

Тогда анализ продолжительности ЖЦ ПО вычисляется по формуле:

$$T_{\text{прогноз}} = f(X_{\text{техн}}, X_{\text{ком}}, X_{\text{ист}}), \quad (2)$$

где $X_{\text{техн}}$ – технические факторы проекта, $X_{\text{ком}}$ – командные факторы проекта, $X_{\text{ист}}$ – исторические факторы проекта, $f(\cdot)$ – функция прогнозирования.

Одним из перспективных направлений повышения эффективности оценки сроков разработки ПО является применение нейронных сетей (НС) и методов машинного обучения [2, 3], которые получили широкое распространение и с помощью которых можно автоматизировать процесс прогнозирования и минимизировать влияние субъективных факторов.

В течение последних десятилетий ведется активная разработка новых подходов для прогнозирования сроков разработки ПО, в том числе с применением нейронных сетей [4, 5]. Однако, часть разработок является прототипом [6, 7], другая часть использует либо устаревшие наборы данных, либо закрытые наборы данных [8], иногда без описания используемых характеристик.

Таким образом, можно отметить, что имеется задача, связанная с повышением эффективности прогнозированием сроков разработки.

Для решения поставленной задачи разработана система прогнозирования сроков разработки [1]. За основу была взята структура, представленная на Рисунке 1.

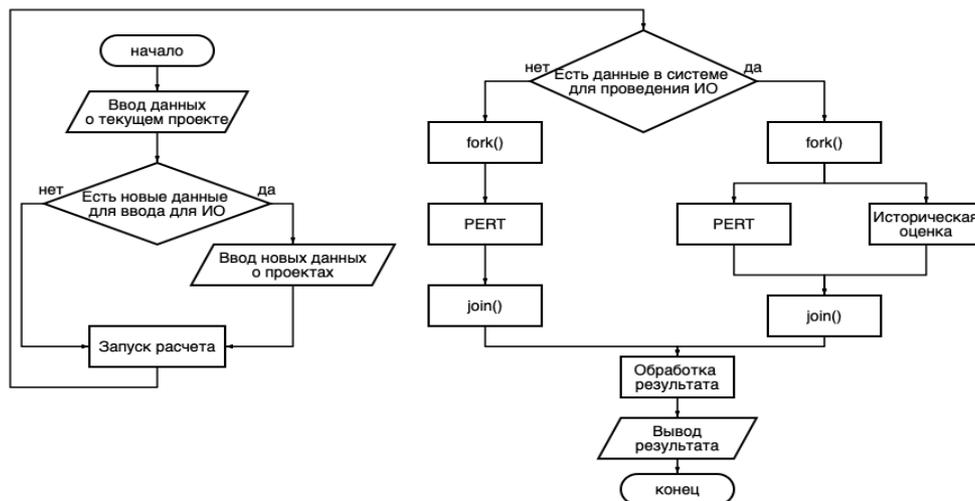


Рисунок 1 – Алгоритм оценки продолжительности разработки программного обеспечения

Figure 1 – Software development duration estimation algorithm

Для обучения нейронных сетей были изучены существующие наборы данных, которые представлены в открытом доступе (Kitchenham, Deshnarian, China и др.). На каждом наборе данных была проведена корреляция для определения признаков целевой переменной (Рисунок 2).

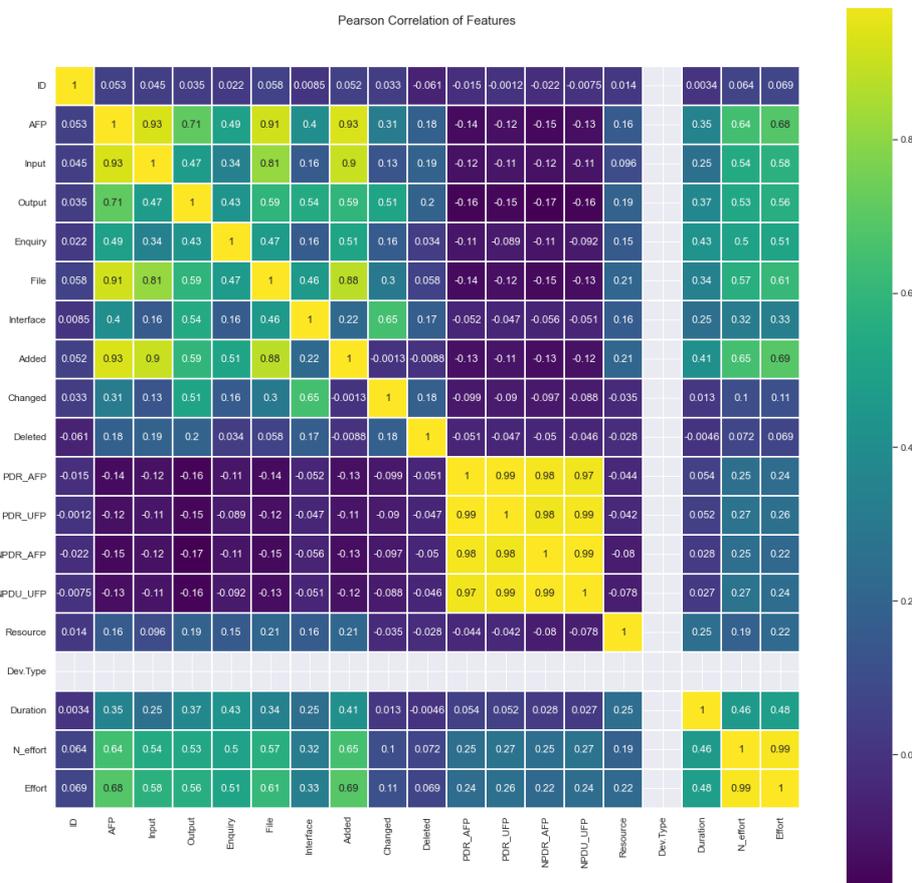


Рисунок 2 – Матрица корреляция переменных
Figure 2 – Variable Correlation Matrix

Целевой (зависимой) переменной в контексте нейронных сетей является признак набора данных, который необходимо предсказать. Для задачи определения сроков разработки ПО целевой переменной является «effort», количество человеко-часов, необходимых на разработку. Признак (независимая переменная) – это набор данных, который влияет на значение целевой переменной. В процессе исследования признаками являлись скорректированные функциональные точки и все элементы из $X_{\text{техн}}$, $X_{\text{ком}}$.

Скорректированные функциональные точки (Adjusted Functional Points, AFP, СФТ) – это нормированная метрика, полученная путем применения весовых коэффициентов или поправочных функций к базовому значению функциональных точек (ФТ, FP) с учетом влияния таких факторов, как, например:

- сложность проекта (техническая, архитектурная, алгоритмическая);
- масштаб проекта (количество модулей, интеграционных точек);
- опыт команды разработчиков;
- требования к надежности и производительности и др.

Множество «Всего задач» из набора данных эквивалентно термину «функциональные точки» (ФТ). При этом, в случае отсутствия данного значения оно вычисляется по формуле:

$$FP = 3B + 2M + S, \quad (3)$$

где B – количество крупных задач, M – количество средних задач, S – количество мелких задач.

Если ФТ – исходное количество функциональных точек, то СФТ можно выразить по формуле:

$$AFP = FP \cdot k_{diff} \cdot k_{scale} \cdot \dots \cdot k_n, \quad (4)$$

где k_{diff} , k_{scale} , k_n – весовые коэффициенты, FP – количество функциональных точек.

Данная формула отличается от международного стандарта ISO/IEC 20926¹ и предлагаемого решения IFPUG, который описывает 5 показателей функциональности и 14 факторов влияния. Однако в Российской Федерации практика показывает, что проекты не расписываются настолько детально, поэтому в данной работе для обучения нейронной сети был использован альтернативный вариант.

Основной причиной изменения формулы (5) [9] являются разнородные открытые наборы данных, в которых «effort» вычисляется разными способами и с разными коэффициентами, либо без публикации формулы вычисления AFP, предоставляя данные «как есть». В работе [10] описана структура входных данных, а также все обозначения и коэффициенты параметров. Пример входных данных для нейронной сети представлен в Таблице 2.

$$AFP = FP \cdot (0,65 + 0,01 \cdot TotalFactor), \quad (5)$$

где FP – необработанные функциональные точки, $TotalFactor$ – сумма значений факторов влияния.

¹ ISO/IEC 20926. Software and Systems Engineering – Software Measurement – IFPUG Functional Size Measurement Method 2009. Geneva: ISO; 2009. 24 p.

Таблица 2 – Пример набора данных для нейронных сетей
Table 2 – Example dataset for a neural networks

| AFP | Effort | AFP2 | AFP3 | Scale |
|------|--------|----------|----------|-------|
| 1587 | 7490 | 1676,81 | 2515,215 | 2 |
| 260 | 4150 | 270,4 | 540,8 | 5 |
| 152 | 668 | 141,7423 | 212,6134 | 2 |
| 252 | 3238 | 280,9911 | 561,9822 | 4 |
| 292 | 2994 | 277,7329 | 555,4658 | 4 |
| 83 | 1333 | 79,18391 | 158,3678 | 5 |
| 79 | 1607 | 81,05195 | 162,1039 | 5 |
| 97 | 1158 | 78,4083 | 156,8166 | 5 |
| 116 | 1243 | 117,0087 | 234,0175 | 4 |

Структура системы для прогнозирования сроков разработки состоит из трех нейронных сетей, моделируя работу алгоритма PERT.

В рамках проведенного исследования проводились эксперименты с разными параметрами нейронных сетей. Основными гиперпараметрами нейронных сетей являются: количество слоев, количество нейронов в каждом слое, типы слоев, функции активации, число эпох, процент исключения нейронов (dropout rate). В ходе экспериментов использовались вариации следующих значений, представленных в Таблице 3.

Таблица 3 – Гиперпараметры нейронных сетей
Table 3 – Hyperparameters of neural networks

| Гиперпараметр | Значения |
|-----------------------------------|-------------------------------|
| Количество слоев | 3–10 |
| Количество нейронов в каждом слое | 16, 32, 64, 128, 256, ... |
| Типы слоев | Полносвязный, рекуррентные |
| Функции активации | Relu, Sigmoid, elu |
| Число эпох | 50, 100, 500, 1000, 1500, ... |
| Dropout rate | 0,1, 0,2, 0,3 |

Для обучения нейронных сетей использовались различные комбинации указанных гиперпараметров для поиска экспериментальным путем наиболее оптимального набора параметров, который гарантирует точный прогноз с минимальными выбросами.

Один из итоговых наборов данных для обучения нейронной сети, прогнозирующий срок разработки ПО, состоит из нескольких комбинации наборов данных, представленных в открытом доступе и приведенных к единой форме, и дополненный практическими данными.

Важно отметить, что во всех представленных наборах данных имеется большой разброс значений человеко-часов. Сильная неоднородность распределения значений приводит к проблемам обучения НС. Для устранения этой проблемы использовалось логарифмическое преобразование по формуле (6). Данная функция позволяет сохранить информацию о небольших проектах.

$$y = \ln(1 + y), \quad (6)$$

где y – количество человеко-часов.

Рассмотрим следующий пример. Допустим, диапазон значений Effort: [120, 24500] человеко-часов. Тогда стандартное отклонение для указанного диапазона $\sigma = 4200$. После преобразования по формуле (6) диапазон Effort сократится до [4,8, 10,1], а стандартное отклонение $\sigma' = 1,2$.

При работе с НС этот подход позволяет ускорить сходимость и уменьшает среднюю квадратичную ошибку.

Предсказание НС \hat{y}' соответствует натуральному логарифму трудозатрат. Для получения физического смысла трудозатрат (человеко-часов) применяется обратное преобразование по формуле:

$$\hat{y} = \exp(\hat{y}') - 1, \quad (7)$$

где \hat{y}' – логарифмированное предсказание НС.

Кроме того, на основе первоначального набора данных были сгруппированы еще несколько дополнительных наборов данных, например, для мелких и крупных проектов. Дополнительные исследования проводились на наборах данных для методологии планирования (устаревшие наборы данных) и на различных значениях коэффициентов). Каждый дополнительный набор данных был убран, и в нем были убраны выбросы.

Список наборов данных можно классифицировать следующим образом:

- универсальный набор;
- по масштабу проекта;
- по методологии разработки;
- по полноте данных.

Характеристики некоторых нейронных сетей приведены в Таблице 4. Сравнительный анализ НС представлен в Таблице 5. Каждый из вариантов нейронной сети был обучен на каждом наборе данных, чтобы выбрать подходящую конфигурацию для определенного типа задач.

Таблица 4 – Пример характеристик нейронных сетей
Table 4 – Example of characteristics of neural networks

| | N1 | N2 | N3 | N4 | N5 | N6 |
|-----------------------------------|--------------|------------------------|------------------|----------------------------|--------------------|----------------|
| Количество слоев | 3 | 5 | 4 | 6 | 5 | 4 |
| Количество нейронов в каждом слое | 128/32/1 | 128/64/32/16/1 | 128/64/32/1 | 256/128/64/32/16/1 | 128/64/32/16/1 | 64/32/16/1 |
| Типы слоев | Полносвязный | Рекуррентные | Рекуррентные | Полносвязный | Полносвязный | Полносвязный |
| Функции активации | Relu, Relu | Relu, Relu, Relu, Relu | Relu, Relu, Relu | Relu, elu, Relu, elu, Relu | elu, elu, elu, elu | elu, elu, elu, |
| Число эпох | 100 | 500 | 1000 | 500 | 100 | 5000 |
| Dropout rate | 0,1 | 0,2 | 0,2 | 0,1 | 0,3 | 0,3 |

Таблица 5 – Сравнительный анализ нейронных сетей
Table 5 – Comparative analysis of neural networks

| | N1 | N2 | N3 | N4 | N5 | N6 |
|-----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Погрешность, % | [1, 15] | [3, 13] | [2, 12] | [2, 9] | [0,01, 5] | [0,1, 7] |
| Среднее время обучения, мин | 2 | 3 | 4 | 5 | 4 | 5 |
| Среднее время оценки, мс | ~1 | ~3 | ~1 | ~7 | ~5 | ~2 |

Прогнозируемая продолжительность ЖЦ ПО методом PERT рассчитывается по формуле:

$$I_{PERT} = \left((NN_s + a_s \cdot \sigma_s) + 4(NN_b + a_b \cdot \sigma_b) + (NN_g + a_g \cdot \sigma_g) \right) / 6, \quad (8)$$

где NN_s – результат вычисления НС для мелких проектов, a_s – параметрический коэффициент НС для мелких проектов, σ_s – стандартное отклонение НС для мелких проектов, NN_b – результат вычисления НС с наилучшим результатом, a_b – параметрический коэффициент НС с наилучшим результатом, σ_b – стандартное отклонение НС с наилучшим результатом, NN_g – результат вычисления НС для крупных проектов, a_g – параметрический коэффициент НС для крупных проектов, σ_g – стандартное отклонение НС для крупных проектов.

Итоговый расчет времени разработки программного обеспечения при наличии данных для исторической оценки вычисляется по формуле:

$$K = \left((I_{HA} + a_{HA} \cdot \sigma_{HA}) + (I_{PERT} + \sigma_{PERT}) \right) / 2, \quad (9)$$

где I_{HA} – результат вычисления исторической оценки, a_{HA} – параметрический коэффициент исторической оценки, σ_{HA} – стандартное отклонение исторической оценки, I_{PERT} – результат вычисления алгоритма PERT, σ_{PERT} – стандартное отклонение метода PERT.

При отсутствии исторических данных расчет выполняется на основе метода PERT по формуле:

$$K = I_{PERT} + \sigma_{PERT}, \quad (10)$$

где I_{PERT} – результат вычисления алгоритма PERT, σ_{PERT} – стандартное отклонение метода PERT.

Результаты

Рассмотрим пример на основе экспериментальных данных завершеного проекта.

На начальном этапе оценки была поставлена задача о разработке системы автоматизированного тестирования фондового рынка X. Команда разработчиков состояла из 4 человек, с опытом команды 2 по таблице X. Масштаб проекта являлся очень большим и сложность проекта была высокая. Количество функциональных точек было равно 721. Входные данные представлены в Таблице 6.

Таблица 6 – Входные данные проекта
Table 6 – Project input data

| ФТ | Масштаб проекта | Сложность проекта | Кол-во разработчиков | Опыт команды |
|-----------|------------------------|--------------------------|-----------------------------|---------------------|
| 721 | 4 | 4 | 4 | 2 |

Аналитик провел краткий анализ и получил результат, что на разработку системы потребуется 190 рабочих дней. На выполнение прогноза эксперту потребовалось около 12 часов. Система поддержки принятия решений (СППР) на основе этих данных выдала прогноз в 239 рабочих дней, а фактический результат разработки составил 243 рабочих дня. Результат представлен в Таблице 7.

Таблица 7 – Результат оценок
Table 7 – Estimated result

| | Фактический результат | Эксперт | Метод анализа |
|---------------------------|------------------------------|----------------|----------------------|
| Количество дней | 243 | 190 | 239 |
| Погрешность оценки | 0 | 53 | 4 |
| Среднеквадратичная ошибка | 0 | 0,08 | 0,0003 |

Таким образом, можно сделать вывод, что рассмотренный в предыдущих главах алгоритм работает точнее, а также скорость выполнения анализа при использовании данного метода повышается в несколько раз.

В ходе апробации были получены результаты, представленные в Таблице 8.

Таблица 8 – Трудозатраты на оценку сроков разработки программного обеспечения
Table 8 – Labor costs for software development time estimation

| Наименование работ | Время выполнения, ч | |
|-----------------------------------|-------------------------------|------------------------------|
| | Без использования СППР | С использованием СППР |
| Анализ проекта | | |
| Сбор исходных данных проекта | 1 | 1 |
| Оформление входных данных проекта | 1 | 0,4 |
| Анализ продолжительности ЖЦ ПО | 10–80 | 0,1 |
| ИТОГО | 12–82 | 1,5 |

В данной таблице представлено сравнение примерных временных затрат при анализе одного и того же проекта экспертом с использованием программы и без нее. Например, по экспертным оценкам, если ранее для анализа сроков разработки одного проекта специалисту требовалось от 12 часов, то с использованием программы за меньшее время (1,5 часа) инженер-аналитик может провести предварительный анализ.

Заключение

В ходе проведенного исследования и разработки была предложена и реализована система анализа продолжительности ЖЦ разработки ПО на основе интеллектуальной обработки информации.

Рассмотренная система позволяет сократить время на анализ и повысить его эффективность за счет использования технологии искусственного интеллекта. Экономия временных ресурсов является существенным конкурентным преимуществом по сравнению с другими алгоритмами и методами анализа продолжительности ЖЦ ПО и достигается средствами автоматизации анализа данных и прикладного применения нейронных сетей.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Шульга Т.Э., Храмов Д.Э. Архитектура программной системы для оценки продолжительности разработки программного обеспечения. *Программная инженерия*. 2024;15(9):476–484. <https://doi.org/10.17587/prin.15.476-484>
Shulga T.E., Khramov D.E. Software System Architecture for Estimating Software Development Time. *Software Engineering*. 2024;15(9):476–484. (In Russ.). <https://doi.org/10.17587/prin.15.476-484>
2. Чжан В., Мухамад И., Саклаков В.М., Джаякоди Д.Н.К. Нейронная сеть для оптимизации адаптивного экспоненциального алгоритма декодирования минимальной суммы. *Моделирование, оптимизация и информационные технологии*. 2025;13(1). (На англ.). <https://doi.org/10.26102/2310-6018/2025.48.1.026>
Zhang W., Mouhamad I., Saklakov V.M., Jayakody D.N.K. Neural Network to Optimize the Adaptive Exponential Min Sum Decoding Algorithm. *Modeling, Optimization and Information Technology*. 2025;13(1). <https://doi.org/10.26102/2310-6018/2025.48.1.026>
3. Гу Чуньюй, Громов М.Л. Искусственная нейронная сеть подавления артефактов наложения изображений для изменения атрибутов лица на основе дифференциальной активации. *Моделирование, оптимизация и информационные технологии*. 2025;13(3). <https://doi.org/10.26102/2310-6018/2025.50.3.013>
Gu Chongyu, Gromov M.L. Artificial Neural Network for Image Blending Artifact Suppression in Differential Activation-Based Face Attribute Editing. *Modeling, Optimization and Information Technology*. 2025;13(3). (In Russ.). <https://doi.org/10.26102/2310-6018/2025.50.3.013>
4. Oba K.M. Development of a Scheffe's Model to Predict the Durations of Project Tasks. *Journal of Engineering Research and Reports*. 2024;26(1):117–124. <https://doi.org/10.9734/JERR/2024/v26i11067>
5. Garcia-Diaz N., Garcia-Virgen Ju., Farias-Mendoza N., et al. Software Development Time Estimation Based on a New Neuro-Fuzzy Approach. In: *2015 10th Iberian Conference on Information Systems and Technologies (CISTI), 17–20 June 2015, Aveiro, Portugal*. IEEE; 2015. P. 1–7. <https://doi.org/10.1109/CISTI.2015.7170378>
6. López-Martín C., Abran A. Neural Networks for Predicting the Duration of New Software Projects. *Journal of Systems and Software*. 2015;101:127–135. <https://doi.org/10.1016/j.jss.2014.12.002>
7. Singal P., Kumari A.Ch., Sharma P. Estimation of Software Development Effort: A Differential Evolution Approach. *Procedia Computer Science*. 2020;167:2643–2652. <https://doi.org/10.1016/j.procs.2020.03.343>
8. Hamada M.A., Abdallah A., Kasem M., Abokhalil M. Neural Network Estimation Model to Optimize Timing and Schedule of Software Projects. In: *2021 IEEE International Conference on Smart Information Systems and Technologies (SIST), 28–30 April 2021, Nur-Sultan, Kazakhstan*. IEEE; 2021. P. 1–7. <https://doi.org/10.1109/SIST50301.2021.9465887>
9. Bundschuh M., Dekkers C. *The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement*. Berlin, Heidelberg: Springer; 2008. 644 p. <https://doi.org/10.1007/978-3-540-68188-5>
10. Храмов Д.Э. Нормализация разнородных наборов данных для прогнозирования сроков разработки программного обеспечения. В сборнике: *Проблемы управления в социально-экономических и технических системах: Материалы XXI Международной научно-практической конференции, 17–18 апреля 2025 года, Саратов, Россия*. Саратов: Издательский центр «Наука»; 2025. С. 202–206.

ИНФОРМАЦИЯ ОБ АВТОРЕ / INFORMATION ABOUT THE AUTHOR

Храмов Дмитрий Эдуардович, аспирант, кафедра «Прикладные информационные технологии», Саратовский государственный технический университет имени Гагарина Ю.А., Саратов, Российская Федерация. **Dmitry E. Khramov**, Postgraduate, Department of Applied Informatics, Yuri Gagarin State Technical University of Saratov, Saratov, the Russian Federation.

Ю.А., Саратов, Российская Федерация.

e-mail: dmitriy-hramov@list.ru

ORCID: [0009-0009-7873-7030](https://orcid.org/0009-0009-7873-7030)

Статья поступила в редакцию 14.08.2025; одобрена после рецензирования 18.09.2025; принята к публикации 27.09.2025.

The article was submitted 14.08.2025; approved after reviewing 18.09.2025; accepted for publication 27.09.2025.