

УДК 681.518

ИНСТРУМЕНТАЛЬНЫЕ ПОДХОДЫ К ОЦЕНКЕ ИНФОРМАЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ В РАМКАХ СТРАТЕГИЧЕСКОГО И ВНУТРИФИРМЕННОГО ПЛАНИРОВАНИЯ ДЕЯТЕЛЬНОСТИ НА ОСНОВЕ PYTHON

Круглов В.Н.,

Калужский филиал Российской академии народного хозяйства и государственной службы при Президенте РФ, Калуга,
email: vladkaluga@yandex.ru, samatova5995@mail.ru

Саматова А.И.,

Калужский филиал Российской академии народного хозяйства и государственной службы при Президенте РФ, Калуга,
email: vladkaluga@yandex.ru, samatova5995@mail.ru

Аннотация. Исследование посвящено разработке инструментальных подходов для оценки информационных систем управления в обрабатывающих производствах с учетом стратегического и внутрифирменного планирования. Основной целью работы является создание программного решения на языке Python, которое интегрирует количественные и качественные параметры оценки, обеспечивая объективный анализ эффективности информационной системы управления. В качестве методологической основы использован метод взвешенной суммы критериев, который позволяет учесть ключевые показатели производительности и удовлетворенности пользователей. Научная новизна заключается в применении экономико-математического моделирования для нормализации и оценки параметров информационной системы управления, включая объем оперативной памяти, частоту работы процессора, время отклика системы и количество операций, выполняемых ею. Разработка инструментов предоставляет предприятиям возможности для адаптации к изменяющимся условиям оптимизации бизнес-процессов и повышения конкурентоспособности.

Ключевые слова: автоматизация процессов, аналитические технологии, инновационные инструменты, программные инструменты, производственная эффективность.

INSTRUMENTAL APPROACHES TO THE ASSESSMENT OF MANAGEMENT INFORMATION SYSTEMS WITHIN THE FRAMEWORK OF STRATEGIC AND INTRA-COMPANY ACTIVITY PLANNING BASED ON PYTHON

Kruglov V.N.,

Kaluga Branch of the Russian Presidential Academy of National Economy and Public Administration, Kaluga,
email: vladkaluga@yandex.ru, samatova5995@mail.ru

Samatova A.I.,

Kaluga Branch of the Russian Presidential Academy of National Economy and Public Administration, Kaluga,
email: vladkaluga@yandex.ru, samatova5995@mail.ru

Abstract. The research is devoted to the development of instrumental approaches for evaluating information management systems in manufacturing industries, taking into account strategic and intra-company planning. The main purpose of the work is to create a software

solution in Python that integrates quantitative and qualitative assessment parameters, providing an objective analysis of the effectiveness of an information management system. The method of weighted sum of criteria is used as a methodological basis, which allows you to take into account key performance indicators and user satisfaction. The scientific novelty lies in the application of economic and mathematical modeling to normalize and evaluate the parameters of an information management system, including the amount of RAM, processor frequency, system response time and the number of operations performed by it. The development of tools provides enterprises with opportunities to adapt to changing conditions for optimizing business processes and increasing competitiveness.

Keywords: process automation, analytical technologies, innovative tools, software tools, production efficiency.

В условиях цифровой трансформации обрабатывающие производства сталкиваются с необходимостью модернизации методов управления. Внутрифирменное и стратегическое планирование становится ключевым инструментом для обеспечения интеграции различных функций предприятия, оптимального распределения ресурсов и повышения конкурентоспособности. Однако стремительное развитие технологий, динамика внешней среды и вызовы технологического суверенитета обуславливают поиск новых подходов к оценке и внедрению информационных систем управления [1].

Информационные системы управления, такие как MRP, MRPII, ERP, PLM, CRM и другие, предлагают широкий спектр функциональных возможностей для оптимизации бизнес-процессов, управления данными и принятия стратегических решений. В то же время их использование сопровождается сложностями, включая высокую стоимость внедрения, требования к обучению персонала и недостаток инструментов, адаптированных к специфическим потребностям предприятий. Данные вызовы требуют разработки инновационных инструментальных подходов для оценки эффективности информационной системы управления на основе внутрифирменного и стратегического планирования. Актуальность исследования определяется противоречиями, возникающими в процессе выбора и использования информационных систем управления:

- необходимость интеграции различных функций и отделов предприятия ограничивается недостаточной адаптивностью существующих решений к внешним изменениям;

- традиционные подходы к планированию, опирающиеся на исторические данные, теряют свою актуальность в условиях высокой изменчивости внешней среды;

- недостаточный доступ к качественным данным ограничивает возможности применения информационных систем управления для оптимизации процессов на основе больших данных и искусственного интеллекта;

- высокие затраты и сложность внедрения информационных систем управления требуют инструментов, которые позволят объективно оценить их эффективность и целесообразность интеграции.

Научная новизна исследования заключается в применении инструментальных подходов, реализованных на базе Python, для оценки информационных систем управления в обрабатывающих производствах. Использование методов программирования для анализа и визуализации данных позволяет интегрировать количествен-

ные и качественные параметры оценки, обеспечивая более глубокий и объективный анализ эффективности информационной системы управления.

Практическая значимость работы состоит в разработке программного продукта, который позволяет предприятиям на основе прозрачных критериев оценивать пригодность и эффективность информационной системы управления в рамках стратегического и внутрифирменного планирования. Инструмент, созданный на базе Python, может быть легко адаптирован под потребности конкретных производств, способствуя оптимизации процессов и снижению затрат.

Исследование направлено на устранения разрыва между современными вызовами управления производственными процессами и существующими подходами к оценке информационной системы управления, предлагая инновационные решения, основанные на интеграции методов программирования и экономико-математического моделирования.

Цель исследования

Цель исследования состоит в разработке инструментальных подходов к оценке информационных систем управления, применимых в рамках стратегического и внутрифирменного планирования деятельности обрабатывающих производств, с использованием языка программирования Python.

Материал и методы исследования

Современные условия ведения бизнеса требуют от предприятий эффективного управления, основанного на комплексном анализе, как внутренних процессов, так и внешних факторов, влияющих на их деятельность. В условиях значительного объема управленческой информации только использование современных информационно-телекоммуникационных технологий позволяет оптимизировать обработку данных и повысить обоснованность принимаемых решений. Информационные системы управления являются ключевым инструментом в обеспечении системного подхода к управлению современными предприятиями. Данные системы объединяют технические, программные и организационные компоненты, а также методы и регламенты, направленные на автоматизацию и оптимизацию процессов сбора, обработки и анализа данных. Их внедрение способствует улучшению качества управления, оперативности предприятия решений и адаптации к изменениям внешней среды. Анализ научных публикаций по теме информационных систем управления выявляет широкий спектр подходов, раскрывающих их значимость для бизнес-процессов и экономической эффективности предприятий. Одним из ключевых направлений исследования является интеграция информационных систем управления в бизнес-процессы, что способствует оптимизации операционной деятельности, сокращению издержек и повышению обоснованности управленческих решений, что позволяет предприятиям эффективно адаптироваться к изменениям внешней среды и укреплять свои позиции на рынке. Важное значение имеет разработка методических подходов и инструментов для оценки эффективности информационной системы управления. Такие подходы учитывают специфику производственной деятельности и обеспечивают возможность проведения комплексного анализа с акцентом на экономические и организационные аспекты, что особенно актуально для обрабатывающей промышленности. Исследования также подчеркивают влияние информационных систем управления на качество обслуживания кли-

ентов и совершенствование бизнес-процессов. Автоматизация процессов улучшает оперативность выполнения заказов, взаимодействие с клиентами и качество услуг, что в конечном итоге способствует укреплению конкурентных позиций предприятий. Особое внимание уделяется финансовым аспектам внедрения информационных систем управления. Автоматизация учета, анализа и отчетности повышает прозрачность и управляемость финансов, снижает риски и способствует эффективному стратегическому планированию. Информационные системы управления выступают важным инструментом, обеспечивающим предприятиям возможности для повышения эффективности, оптимизации ресурсов и улучшения взаимодействия с клиентами и партнерами. Внутрифирменное и стратегическое планирование обрабатывающих производств требует осознанного выбора и тщательной интеграции информационных систем управления – MRP, MRPII, MES, IDM, PDM, SCADA, ECM, EAM, MDM, WMS, ERP, PLM, CRM, BIM, SCM, APS, BPM, DSS, PIM, DLP, SRM, BI, представляют собой комплекс технологий и инструментов, направленных на оптимизацию процессов, управление ресурсами и повышение производительности. Их применение позволяет предприятиям достигать стратегических и операционных целей, обеспечивая конкурентные преимущества в условиях динамичной экономической среды [2].

Для оптимизации расчетов экономико-математической модели оценки информационной системы управления в обрабатывающем производстве предлагается разработка программного решения с использованием языка программирования. При выборе языка программирования необходимо учитывать специфику проекта и доступные инструменты. Например, такие языки, как PHP, специализируются на веб-разработке, тогда как C++ и C# являются универсальными инструментами для сложных задач, включая управление трехмерными объектами. Однако для разработки программного обеспечения, ориентированного на экономико-математические расчеты, Python является предпочтительным выбором благодаря своей простоте и освоении, функциональной гибкости и широкому набору библиотек [3-5].

Python предоставляет пользователям интуитивно понятную среду программирования с мощными инструментами для обработки данных и выполнения сложных вычислений. Его основное преимущество заключается в удобстве отладки кода: ошибки быстро выявляются и корректируются, что ускоряет процесс разработки. Кроме того, множество доступных библиотек Python позволяют решать задачи математического моделирования, анализа данных, визуализации и автоматизации. Для реализации экономико-математической модели предлагается использовать PyCharm: Community Edition – интегрированную среду разработки, разработанную JetBrains. Данная среда предоставляет широкий набор функций, включая:

- интеллектуальное завершение кода, которое ускоряет написание программ и снижает вероятность ошибок;
- встроенный отладчик, упрощающий диагностику и устранение проблем через точки останова и мониторинг переменных;
- мощные инструменты рефакторинга, повышающие читаемость и структурированность кода;
- интеграцию с системами контроля версий, облегчающую отслеживание изменений и совместную работу над проектами;
- анализатор кода, выявляющий потенциальные ошибки и предлагающий улучшения для повышения эффективности.

PyCharm: Community Edition бесплатная и подходит для проектов, использующих стандартную библиотеку Python. Последняя включает широкий набор встроенных модулей для выполнения математических операций, обработки статистических данных, работы с датами, строками, генерации случайных чисел и управления сложными структурами данных. Данные возможности делают PyCharm: Community Edition идеальными инструментами для реализации экономико-математической модели, обеспечивая интеграцию с набором данных и генерацию результатов, соответствующих специфике задачи [6-7]. Использование Python не только ускоряет процесс разработки, но и обеспечивает высокую точность расчетов, гибкость решений и возможность их масштабирования под задачи стратегического и внутрифирменного планирования в обрабатывающих производствах.

Результаты и их обсуждение

Для оценки информационной системы управления как инструмента внутрифирменного и стратегического планирования деятельности обрабатывающего производства применяется метод взвешенной суммы критериев (МВСК) [8-10]. Основным результатом использования данного метода является обобщенный показатель оценки информационной системы управления в обрабатывающем производстве – $O_{\text{СУОП}}$. Данный показатель интегрирует как количественные, так и качественные характеристики, что позволяет формировать объективную оценку эффективности информационных систем управления. В основу $O_{\text{СУОП}}$ заложен комплексный подход, которые учитывает количественные параметры, такие как производительность системы управления, и качественные показатели, включая уровень удовлетворенности пользователей системой. Такой подход обеспечивает сбалансированную оценку функциональности и эффективности информационной системы управления.

Математически формула МВСК для $O_{\text{СУОП}}$ выражается следующим образом:

$$O_{\text{СУОП}} = w_{\text{ООП}} * \text{ООП} + w_{\text{ЗОСУ}} * \text{ЗОСУ} + w_{\text{ЧРЦП}} * \text{ЧРЦП} + w_{\text{КОВСУ}} * \text{КОВСУ} + w_{\text{УСУ}} * \text{УСУ} \quad (2.1)$$

где ООП – объем оперативной памяти, ЗОСУ – задержка отклика системы управления, ЧРЦП – частота работы центрального процессора, КОВСУ – количество операций выполняемых системой управления, УСУ – удовлетворенность системой управления, взвешенных коэффициентами их относительной важности – w_i : $w_{\text{ООП}}$ – отражает важность объема оперативной памяти для работы информационной системы управления, более высокий коэффициент означает, что данный параметр играет критическую роль в оценке системы; $w_{\text{ЗОСУ}}$ – указывает на значимость времени отклика системы, минимизация задержек повышает производительность информационной системы управления, что делает этот показатель важным в высоконагруженных системах; $w_{\text{ЧРЦП}}$ – отображает вычислительные возможности системы, где более высокая частота процессора обеспечивает ускорение выполнения операций; $w_{\text{КОВСУ}}$ – характеризует объем выполняемых операций, данный параметр показывает степень загруженности и масштаб функциональности информационной системы управления; $w_{\text{УСУ}}$ – отражает удовлетворенность пользователей системой, высокий коэффициент указывает на значимость пользовательского опыта, включая удобство, функциональность и общую производительность.

Для расчета экономико-математической модели оценки информационных систем управления в обрабатывающем производстве воспользуемся функцией `calculatate_final_result` на Python в PyCharm: Community Edition (рис. 1).

```
# общая формула ОСУОП = wооп * ооп + wзосу * зосу + wчрцп * чрцп + wковсу * ковсу + wусу * усу
def calculate_final_result(vectors: list, OOP, ZOSU, CHRCP, KOVSU, USU) -> float:
    return OOP * vectors[0] + ZOSU * vectors[1] + CHRCP * vectors[2] + KOVSU * vectors[3] + USU *
    vectors[4]
```

Рис. 1. Расчет окончательного результата ОСУОП

Согласно рисунку 1, строка `def calculate_final_result(vectors: list, OOP, ZOSU, CHRCP, KOVSU, USU)` определяет функцию с именем `calculate_final_result`, которая принимает параметры `vectors` – ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ, где `vectors` являются коэффициенты важности для каждого критерия – $w_{\text{ооп}}$, $w_{\text{зосу}}$, $w_{\text{чрцп}}$, $w_{\text{ковсу}}$, $w_{\text{усу}}$ и возвращает число с плавающей запятой через `float`. Тип `list` используется для представления параметров векторов. Строка `return OOP * vectors[0] + ZOSU * vectors[1] + CHRCP * vectors[2] + KOVSU * vectors[3] + USU * vectors[4]` вычисляет окончательный результат на основе предоставленных критериев и векторов ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ возвращает результат расчета, с помощью функции `return`. Далее, осуществим нормализацию критериев ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ на Python в PyCharm: Community Edition (рис. 2).

```
# Нормализованные константы
OOP = 0 if OOP_INPUT < OOP_THRESHOLD else 1
ZOSU = 0 if ZOSU_INPUT > ZOSU_THRESHOLD else 1
CHRCP = 0 if CHRCP_INPUT > CHRCP_THRESHOLD else 1
KOVSU = 0 if KOVSU_INPUT > KOVSU_THRESHOLD else 1
```

Рис. 2. Нормализация критериев ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ на Python в PyCharm

Согласно рисунку 2, в коде представленные входные пороговые значения `OOP_INPUT`, `ZOSU_INPUT`, `CHRCP_INPUT`, `KOVSU_INPUT`, относительно которых соответствующие входные значения `OOP_THRESHOLD`, `ZOSU_THRESHOLD`, `CHRCP_THRESHOLD`, `KOVSU_THRESHOLD` сравниваются для определения значений нормализованных критериев ООП, ЗОСУ, ЧРЦП, КОВСУ через логических операторов `if` и `else`. `If` используется для проверки условия, если условие оценивается как истинное, то выполняется блок кода, следующий за логическим оператором `if`. `Else` используется вместе с `if` для выполнения блока кода, если условие `if` принимает ложное значение. Если входная переменная ниже соответствующего порога, соответствующему критерию присваивается значение 0, в противном случае ей присваивается значение 1.

Введем условные значения для `OOP_THRESHOLD`, `ZOSU_THRESHOLD`, `CHRCP_THRESHOLD`, `KOVSU_THRESHOLD` (рис. 3) и для `OOP_INPUT`, `ZOSU_INPUT`, `CHRCP_INPUT`, `KOVSU_INPUT` (рис. 4), чтобы в дальнейшем проверить функциональность и работоспособность кода.

На рисунке 3 представлены условные значения, присвоенные пороговым критериям необходимы для нормализации критериев экономико-математической модели. Каждый порог соответствует определенной переменной `OOP_THRESHOLD` – пороговое значение объема оперативной памяти (ООП), `ZOSU_THRESHOLD` – пороговое значение задержки отклика системы управления (ЗОСУ), `CHRCP_THRESHOLD` – пороговое значение частоты работы центрального процессора (ЧРЦП), `KOVSU_THRESHOLD` – пороговое значение количества операций, выпол-

няемых системой управления (КОВСУ), $USU_THRESHOLD$ – пороговое значение удовлетворенности системой управления (УСУ). Данные пороговые значения действуют как контрольные точки, с которыми будут сравниваться фактические входные значения для нормализации критериев.

Пограничные значения для нормализации констант

$OOIP_THRESHOLD = 119$
 $ZOSU_THRESHOLD = 0.02$
 $CHRCP_THRESHOLD = 6$
 $KOVSU_THRESHOLD = 4$
 $USU_THRESHOLD = 10$

Рис. 3. Условные значения для входных пороговых значений $OOIP_THRESHOLD$, $ZOSU_THRESHOLD$, $CHRCP_THRESHOLD$, $KOVSU_THRESHOLD$

Константы по запросу у предприятия (для финального расчета)

$OOIP_INPUT = 250$
 $ZOSU_INPUT = 0.025$
 $CHRCP_INPUT = 4$
 $KOVSU_INPUT = 1.7$

Рис. 4. Условные значения для входных пороговых значений $OOIP_INPUT$, $ZOSU_INPUT$, $CHRCP_INPUT$, $KOVSU_INPUT$

На рисунке 4 представлены условные значения входных данных фактических показателей информационной системы управления, сформированные для тестирования функциональности кода, которые будут сравниваться с ранее определенными пороговыми значениями из рисунка 3. Код использует эти входные данные для расчета и нормализации определённых критериев в экономико-математической модели, способствуя общей оценки информационных систем управления на обрабатывающем производстве. Далее осуществим расчет матриц ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ для выявления значений каждого критерия коэффициента важности W_{oop} , W_{zosu} , W_{chrcp} , W_{kovsu} , W_{usu} (рис. 5).

На рисунке 5 представлено несколько функций, используемых для расчета матриц ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ, данные функции предназначены для генерации матриц на основе определенных критериев и коэффициентов.

Матрицы рассчитываются на основе входных коэффициентов, например, A_{ZOSU_OOIP} , A_{OOIP_CHRCP} и других, а также соответствующих обратных значений. Функция $calc_matrix_kv(matrix: list) \rightarrow list$: вычисляет среднегеометрическое значение строк в матрице и возвращает список нормализованных значений на основе средних геометрических каждой строки. В качестве входных данных принимается список $matrix$, представляющий матрицу, для которой необходимо вычислить среднее геометрическое значение.

Функция перебирает каждую строку матрицы, вычисляет среднее геометрическое значение в каждой строке и добавляет результат в список $rows_geom_averages$.

Далее вычисляется сумма средних геометрических значений всех строк $matrix_geom_average$ и возвращается список нормализованных значений, полученных путем деления каждого среднегеометрического значения на сумму, округленную до двух десятичных знаков.

```

from input values import *

def calc_matrix_k(matrix: list) -> list:
    matrix_geom_average = 0
    rows_geom_averages = []
    for i in range(0, len(matrix)):
        buff = 1
        for j in range(0, len(matrix[i])):
            buff *= matrix[j][i]
        buff = buff ** (1 / (len(matrix[i])))
        rows_geom_averages.append(buff)
        matrix_geom_average += buff
    return [round(i / matrix_geom_average, 2) for i in rows_geom_averages]

def oop_matrix_calculator() -> list:
    return [
        [1, 1/A_ZOSU_OOP, 1/A_CHRCP_OOP, 1/A_KOVSU_OOP, 1/A_USU_OOP],
        [A_OOP_ZOSU, 1, A_CHRCP_ZOSU, A_KOVSU_ZOSU, A_USU_ZOSU],
        [A_OOP_CHRCP, 1/A_ZOSU_CHRCP, 1, 1/A_KOVSU_CHRCP, 1/A_USU_CHRCP],
        [A_OOP_KOVSU, 1/A_ZOSU_KOVSU, A_CHRCP_KOVSU, 1, 1/A_USU_KOVSU],
        [A_OOP_USU, 1/A_ZOSU_USU, A_CHRCP_USU, A_KOVSU_USU, 1],
    ]

def zosu_matrix_calculator() -> list:
    return [
        [1, A_OOP_ZOSU, 1/A_CHRCP_ZOSU, 1/A_KOVSU_ZOSU, A_USU_ZOSU],
        [1/A_ZOSU_OOP, 1, 1/A_CHRCP_OOP, 1/A_KOVSU_OOP, A_USU_OOP],
        [A_ZOSU_CHRCP, A_OOP_CHRCP, 1, A_KOVSU_CHRCP, A_USU_CHRCP],
        [A_ZOSU_KOVSU, A_OOP_KOVSU, 1/A_CHRCP_KOVSU, 1, 1/A_USU_KOVSU],
        [A_ZOSU_USU, 1/A_OOP_USU, 1/A_CHRCP_USU, A_KOVSU_USU, 1],
    ]

def chrpc_matrix_calculator() -> list:
    return [
        [1, A_OOP_CHRCP, A_ZOSU_CHRCP, 1/A_KOVSU_CHRCP, 1/A_USU_CHRCP],
        [A_CHRCP_OOP, 1, 1/A_ZOSU_OOP, 1/A_KOVSU_OOP, 1/A_USU_OOP],
        [1/A_CHRCP_ZOSU, A_OOP_ZOSU, 1, A_KOVSU_ZOSU, 1/A_USU_ZOSU],
        [A_CHRCP_KOVSU, A_OOP_KOVSU, 1/A_ZOSU_KOVSU, 1, 1/A_USU_KOVSU],
        [A_CHRCP_USU, A_OOP_USU, 1/A_ZOSU_USU, A_KOVSU_USU, 1],
    ]

def kovsu_matrix_calculator() -> list:
    return [
        [1, 1/A_OOP_KOVSU, 1/A_ZOSU_KOVSU, A_CHRCP_KOVSU, 1/A_USU_KOVSU],
        [A_KOVSU_OOP, 1, A_ZOSU_OOP, A_CHRCP_OOP, A_USU_OOP],
        [A_KOVSU_ZOSU, 1/A_OOP_ZOSU, 1, 1/A_CHRCP_ZOSU, A_USU_ZOSU],
        [1/A_KOVSU_CHRCP, 1/A_OOP_CHRCP, A_ZOSU_CHRCP, 1, 1/A_USU_CHRCP],
        [A_KOVSU_USU, 1/A_OOP_USU, A_ZOSU_USU, A_CHRCP_USU, 1],
    ]

def usu_matrix_calculator() -> list:
    return [
        [1, A_OOP_USU, A_ZOSU_USU, A_CHRCP_USU, A_KOVSU_USU],
        [1/A_USU_OOP, 1, 1/A_ZOSU_OOP, 1/A_CHRCP_OOP, A_KOVSU_OOP],
        [1/A_USU_ZOSU, A_OOP_ZOSU, 1, 1/A_CHRCP_ZOSU, A_KOVSU_ZOSU],
        [1/A_USU_CHRCP, A_OOP_CHRCP, 1/A_ZOSU_CHRCP, 1, A_KOVSU_CHRCP],
        [1/A_USU_KOVSU, A_OOP_KOVSU, 1/A_ZOSU_KOVSU, A_CHRCP_KOVSU, 1],
    ]

```

Рис. 5. Расчет матриц ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ

```

matrixs = [
    oop_matrix_calculator(),
    zosu_matrix_calculator(),
    chrcp_matrix_calculator(),
    kovsu_matrix_calculator(),
    usu_matrix_calculator()
]
supermatrix = []
for matrix in matrixs:
    supermatrix.append(calc_matrix_kv(matrix))

print('МАТРИЦЫ\n\n')
pprint(matrixs)
print('_____\n\n')
print('Суперматрица (векторы для каждой матрицы)\n\n')
pprint(supermatrix)
print('_____\n\n')

print('Расчет векторов суперматрицы\n\n')
supermatrix_vectors = calc_matrix_kv(supermatrix)
pprint(supermatrix_vectors)
print('_____\n\n')

```

Рис. 6. Расчет матриц и суперматриц ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ

Необходимо продолжить расчет матриц и рассчитать суперматрицу (рис. 6).

На рисунке 6, матрицы представляющие различные критерии ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ, рассчитываются с помощью специальных функций, например, oop_matrix_calculator и других, которые возвращают матрицы на основе заранее определенных коэффициентов и критериев. Суперматрица создается путем применения функции calc_matrix_kv(matrix) в каждой матрице, сгенерированной на предыдущем шаге. Суперматрица представляет собой список нормализованных векторов, полученных из среднегеометрических значений строк каждой матрицы. Далее осуществляется распечатка матриц и суперматриц для визуализации, с помощью функции pprint (pretty print).

Функция pprint используется для форматирования вывода для лучшей читаемости кода. Печатные данные включают в себя исходные матрицы, суперматрицу и векторы (коэффициенты важности) для каждой матрицы внутри суперматрицы. Расчет векторов суперматрицы включает в себя вычисление векторов суперматрицы с помощью функции calc_matrix_kv(matrix). Результат supermatrix_vectors представляет собой значения, полученные из среднегеометрических строк суперматрицы. Полученный результат позволяет сформировать представление о взаимосвязях и влиянии между различными критериями экономико-математической модели.

Введем условные значения экспертных оценок важности критерия ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ (рис. 7), чтобы в дальнейшем проверить функциональность и работоспособность кода.

На рисунке 7 представлены условные значения оценки, которые присваиваются экспертами на основе воспринимаемой важности каждого критерия по сравнению

с другими. Шкала показывает воспринимаемую значимость каждого критерия, чем выше присвоенное значение, тем больше воспринимается важность критерия при оценке информационных систем управления на обрабатывающем производстве. Далее необходимо рассчитать УСУ на Python в PyCharm: Community Edition (рис. 8):

На рисунке 8, код импортирует модуль pprint для печати структур данных и модуль statistics для статистических вычислений, два модуля из пользовательских файлов: модуль questions, который включает в себя список вопросов и модуль matrix_functions, который содержит данные рисунка 5, которые необходимы для вычислений. Функция usu_calculator принимает два аргумента questions и answers, перебирает каждый блок вопросов в структуре данных questions. Внутри каждого блока вычисляется медиана ответов экспертов на каждый вопрос с помощью функции statistics.median и сохраняет медианы в block_buff_for_average. Далее вычисляется среднее значение данных медиан для каждого блока и добавляет среднее значение к all_blocks_average_values, возвращает округленную сумму всех средних значений по блокам как общий уровень удовлетворенности УСУ.

```
# аООПЗОСУ, аООПЧРЦП, аООПКОВСУ, аООПУСУ | значение строго из перечня (1, 3, 5, 7, 9)
# Экспертные оценки важности критерия
A_OOP_OOP = 1
A_OOP_ZOSU = 3
A_OOP_CHRCP = 5
A_OOP_KOVSU = 7
A_OOP_USU = 9

A_ZOSU_ZOSU = 1
A_ZOSU_OOP = 3
A_ZOSU_CHRCP = 7
A_ZOSU_KOVSU = 9
A_ZOSU_USU = 5

A_CHRCP_CHRCP = 1
A_CHRCP_OOP = 5
A_CHRCP_ZOSU = 7
A_CHRCP_KOVSU = 3
A_CHRCP_USU = 3

A_KOVSU_KOVSU = 1
A_KOVSU_OOP = 7
A_KOVSU_ZOSU = 9
A_KOVSU_CHRCP = 3
A_KOVSU_USU = 5

A_USU_USU = 1
A_USU_OOP = 9
A_USU_ZOSU = 5
A_USU_CHRCP = 3
A_USU_KOVSU = 5
```

Рис. 7. Условные значения экспертных оценок важности критерия ООП, ЗОСУ, ЧРЦП, КОВСУ, УСУ

```

from pprint import pprint
import statistics

import questions
from matrix.functions import *

# УСУ - удовлетворенность системой управления
def usu_calculator(questions, answers) -> float:
    all_blocks_average_values = []
    for question_block in questions:
        block_buff_for_average = []
        for index, question in enumerate(questions[question_block]):
            answers_buff_for_median = []
            for experts in answers[question_block]:
                expert_answer = experts[index]
                answers_buff_for_median.append(expert_answer)
            block_buff_for_average.append(statistics.median(answers_buff_for_median))
        all_blocks_average_values.append(sum(block_buff_for_average) / len(block_buff_for_average))
    return round(sum(all_blocks_average_values), 2)

if __name__ == '__main__':
    USU = usu_calculator(questions=questions.questions, answers=answers)
    USU = 0 if USU < USU_THRESHOLD else 1

```

Рис. 8. Расчет УСУ на Python в PyCharm: Community Edition

```

# Ответы экспертов по каждому из вопросов каждого блока (пример для двух экспертов)
answers = {
    'Общая удовлетворенность системой (ОУС)': [[1, 2, 5], [4, 3, 5]],
    'Внутрифирменное планирование (ВП)': [[1, 2, 5], [4, 3, 5]],
    'Стратегическое планирование (СП)': [[1, 2, 5], [4, 3, 5]],
    'Опыт пользователя (ОП)': [[1, 2, 5], [4, 3, 5]],
    'Поддержка и обучение (ПиО)': [[1, 2, 5], [4, 3, 5]],
    'Улучшение и обратная связь (УиОС)': [[1, 2, 5], [4, 3, 5]],
}

```

Рис. 9. Условные значения экспертных оценок по блокам ОУС, ВП, СП, ОП, ПиО, УиОС

Строка `if __name__ == '__main__':` вызывает функцию `usu_calculator` с заданными вопросами и ответами. Полученное значение `USU` присваивается переменной `USU`, выполняется проверка порога, если вычисленное значение `USU` меньше заданного порога `USU_THRESHOLD`, значение `УСУ` устанавливается равным 0, в противном случае ему присваивается значение 1. Введем условные значения для расчета ОУС, ВП, СП, ОП, ПиО, УиОС ответов экспертов по каждому вопросу фактора блока каждого блока на примере данных от двух экспертов (рис. 9), чтобы в дальнейшем проверить функциональность и работоспособность кода, связанного с расчетом уровня удовлетворенности на основе экспертных оценок.

На рисунке 9 оценки представлены с использованием числовых значений по шкале Лайкерта, отражающих оценку, присвоенную условными экспертами по каж-

дому вопросу фактора блока в соответствующем блоке. Далее для получения итогового значения оценки информационных систем управления в обрабатывающем производстве на Python в PyCharm: Community Edition (рис. 10).

```
buff = 0
for kv in supermatrix_vectors:
    buff += kv

final_result = calculatate_final_result(supermatrix_vectors, OOP, ZOSU, CHRCP, KOVSU, USU)
print(final_result)
```

Рис. 10. Расчет окончательного результата ОСУОП на Python в PyCharm: Community Edition

На рисунке 10 `buff = 0` инициализирует переменную `buff` с начальным значением 0, данная переменная будет использоваться для накопления суммы коэффициентов важности в `supermatrix_vectors`. Вычисление суммы коэффициентов важности в суперматрице осуществляется через `for kv in supermatrix_vectors: buff += kv`, перебирая каждый элемент `kv` в списке `supermatrix_vectors` и добавляя его значение в переменное `buff`. Данный цикл вычисляет сумму коэффициентов важности в `supermatrix_vectors`. Расчет окончательного результата осуществляется через `final_result = calculatate_final_result(supermatrix_vectors, OOP, ZOSU, CHRCP, KOVSU, USU)`, вызывая функцию `calculatate_final_result` с определенными показателями `supermatrix_vectors, OOP, ZOSU, CHRCP, KOVSU, USU`. Переменная `final_result` сохраняет результат функции `calculatate_final_result`. `Print(final_result)` выводит значение `final_result` на консоль, рассчитывая результат экономико-математической модели ОСУОП. Функциональность и работоспособность кода представлена на рисунке 11, благодаря приведенным условным значениям для расчета экономико-математической модели оценки информационных систему управления в обрабатывающем производстве.

Как видно из рисунка 11, код рассчитывает матрицы, представляющие отношения между различными критериями в модели оценки, векторы суперматриц получаются из матриц для облегчения дальнейших вычислений, программ вычисляет окончательный результата оценки, который в данном случае обозначается как $O_{\text{СУОП}}$, распечатывает значение $O_{\text{СУОП}}$, представляющее собой оценку эффективности системы управления как инструмента внутрифирменного и стратегического планирования деятельности обрабатывающего производства. Полученный результат 0,79 $O_{\text{СУОП}}$ соответствует категории С – удовлетворительное состояние информационных системы управления, что демонстрирует функциональность и работоспособность кода для расчета экономико-математической модели $O_{\text{СУОП}}$. Разработанное программное решение на языке Python реализованное с помощью PyCharm: Community Edition, оптимизирует расчет экономико-математической модели оценки информационных систем управления на обрабатывающем производстве.

Выводы

В рамках исследования были разработаны инструментальные подходы для оценки информационных систем управления, применимые в стратегическом и внутрифирменном планировании обрабатывающих производств. Реализация эконо-

номико-математической модели на языке программирования Python позволила интегрировать количественные и качественные параметры оценки, что обеспечивает всесторонний анализ эффективности информационной системы управления.

C:\Users\sc\PycharmProjects\pythonProject\venv\Scripts\python.exe

C:\Users\sc\PycharmProjects\pythonProject\main.py

МАТРИЦЫ

```
[[[1, 0.3333333333333333, 0.2, 0.14285714285714285, 0.1111111111111111],
 [3, 1, 7, 9, 5],
 [5, 0.14285714285714285, 1, 0.3333333333333333, 0.3333333333333333],
 [7, 0.1111111111111111, 3, 1, 0.2],
 [9, 0.2, 3, 5, 1]],
 [[1, 3, 0.14285714285714285, 0.1111111111111111, 5],
 [0.3333333333333333, 1, 0.2, 0.14285714285714285, 9],
 [7, 5, 1, 3, 3],
 [9, 7, 0.3333333333333333, 1, 0.2],
 [5, 0.1111111111111111, 0.3333333333333333, 5, 1]],
 [[1, 5, 7, 0.3333333333333333, 0.3333333333333333],
 [5, 1, 0.3333333333333333, 0.14285714285714285, 0.1111111111111111],
 [0.14285714285714285, 3, 1, 9, 0.2],
 [3, 7, 0.1111111111111111, 1, 0.2],
 [3, 9, 0.2, 5, 1]],
 [[1, 0.14285714285714285, 0.1111111111111111, 3, 0.2],
 [7, 1, 3, 5, 9],
 [9, 0.3333333333333333, 1, 0.14285714285714285, 5],
 [0.3333333333333333, 0.2, 7, 1, 0.3333333333333333],
 [5, 0.1111111111111111, 5, 3, 1]],
 [[1, 9, 5, 3, 5],
 [0.1111111111111111, 1, 0.3333333333333333, 0.2, 7],
 [0.2, 3, 1, 0.14285714285714285, 9],
 [0.3333333333333333, 5, 0.14285714285714285, 1, 3],
 [0.2, 7, 0.1111111111111111, 3, 1]]]
```

Суперматрица (векторы для каждой матрицы)

```
[[0.52, 0.03, 0.22, 0.15, 0.07],
 [0.35, 0.23, 0.04, 0.1, 0.27],
 [0.2, 0.53, 0.08, 0.16, 0.04],
 [0.36, 0.04, 0.23, 0.2, 0.17],
 [0.03, 0.42, 0.05, 0.08, 0.42]]
```

Расчет векторов суперматрицы

```
[0.29, 0.2, 0.13, 0.18, 0.19]
```

0.79

Process finished with **exit** code 0

Рис. 11. Функциональность и работоспособность кода расчета ОСУОП на Python в PyCharm: Community Edition

Разработанная методика на основе метода взвешенной суммы критериев продемонстрировала высокую точность и адаптивность оценки. Применение Python, благодаря его функциональной гибкости и библиотечным возможностям, позволило реализовать решение, которое:

- упрощает нормализацию данных и расчет весовых коэффициентов критериев;
- автоматизирует процесс построения матриц взаимосвязей и суперматрицы;
- обеспечивает прозрачность и наглядность визуализации результатов.

Полученные результаты могут быть применены как для оценки текущих систем управления, так и для обоснования выбора новых решений, что особенно важно в условиях глобальных вызовов, таких как динамичность внешней среды и необходимость технологического суверенитета.

Литература

1. Абашева О.Ю., Азарян Е.М., Бессарабов В.О. и др. Социально-экономические императивы устойчивого развития России: новый характер экономических взаимосвязей. Самара: ООО «Поволжская научная корпорация» (НИЦ «ПНК»), 2024. 230 с.
2. Круглов В.Н., Саматова А.И. Эволюция планирования в обрабатывающем производстве: от традиционных методов к современным информационным системам управления // Финансовый менеджмент. 2024. № 9. С. 216-226.
3. Гибадуллин Р.Ф., Белашова Е.С. Параллельное программирование на языках C/C++ и C#. Казань: ООО «Редакционно-издательский центр «Школа», 2021. 104 с.
4. Федоров Д.Ю. Программирование на языке высокого уровня Python: учебное пособие. 4-е изд., пер. и доп. М.: Издательство Юрайт, 2023. 214 с.
5. Таршхоева Ж.Т. Язык программирования Python. Библиотеки Python // Молодой ученый. 2021. № 5 (347). С. 20-21.
6. Официальный сайт PyCharm. [Электронный ресурс]. URL: <https://www.jetbrains.com/pycharm/buy/?section=commercial&billing=yearly> (дата обращения 26.10.2024).
7. Samatova A.I. Tools for evaluating information management systems in manufacturing // Научные достижения и инновационные подходы: теория, методология, практика: Сборник научных трудов по материалам XXIII Международной научно-практической конференции (г.-к. Анапа, 29 апреля 2024 г.). Анапа: Изд-во «НИЦ ЭСП» в ЮФО, 2024. С. 58-62.
8. Елисеева И.И. и др. Эконометрика: учебник для вузов / под редакцией И. И. Елисевой. М.: Издательство Юрайт, 2023. 449 с.
9. Гармаш А.Н., Орлова И.В., Федосеев В.В. Экономико-математические методы и прикладные модели: учебник для бакалавриата и магистратуры / под редакцией В. В. Федосеева. 4-е изд., перераб. и доп. М.: Издательство Юрайт, 2022. 328 с.
10. Красс М.С. Математика в экономике. Базовый курс: учебник для вузов. 2-е изд., испр. и доп. М.: Издательство Юрайт, 2024. 470 с.