Прогнозирование технического долга информационных систем при развитии архитектуры предприятия

Мамедгулиев Р.И.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

E-mail: mamedguliev-ruslan2000@yandex.ru

Цуканова О.А.

Санкт-Петербургский государственный университет

E-mail: zoa1999@mail.ru

Аннотация. **В данной статье рассмотрена проблема по накоплению технического долга информационных систем в условиях развития архитектуры предприятия. Целью исследования является разработка методики по оценке и прогнозированию технического долга с учётом характеристик ИТ-ландшафта, а также организационных факторов. В статье предложен комплексный подход, который в том числе предусматривает разграничение ресурсной, бизнес-ориентированной и технической составляющих. Оценка технического долга происходит за счёт общего показателя критичности и масштаба архитектурных нарушений, ресурсная составляющая характеризует трудоёмкость их устранения, а коэффициент бизнес-критичности применяется для определения приоритетов модернизации. Для прогноза технического долга было предложено применение метода случайного леса (Random Forest) на основе архитектурных признаков программных систем. Возможности применения методики проиллюстрированы на примере обезличенных проектов, связанных с заменой информационной системы, переходом к микросервисной архитектуре и развитием в информационных системах интеграционных механизмов. Предложенный подход предназначен для информационной поддержки принятия архитектурных решений и расчётов по распределению разных ресурсов предприятия.**

Ключевые слова: архитектура предприятия; архитектурные характеристики; информационные системы; машинное обучение; прогнозирование; случайный лес; технический долг; управление техническим долгом

I. Введение

В условиях цифровой трансформации организаций возрастает роль архитектуры предприятия как способа обеспечения согласованного развития бизнес-процессов, информационных систем и инфраструктуры технологий. Усложнение ИТ-ландшафта, рост числа интеграционных взаимодействий и необходимость в модели изменения информационных систем сопровождаются созданием архитектурных ограничений гибкости и адаптивности организации. Важным фактором, влияющим на итог изменений в архитектуре, становится накопление технического долга, обусловленного использованием неоптимальных и иных технических решений, устареванием технологий, а также недостаточной согласованностью их компонентов.

Технический долг представляет собой совокупность накопленных ограничений и недостатков программных и архитектурных решений, устранение которых требует ресурсов при сопровождении, модернизации и развитии информационных систем. Возникновение этого фактора связано с необходимостью ускоренного внедрения всех функциональных изменений и слабым вниманием к вопросам сопровождаемости, совместимости интеграций и устойчивости технологий. Накопление технического долга способствует увеличению трудоёмкости проекта разработки, усложнению связей между программными системами и возникновению дополнительных затрат на реализацию архитектурных преобразований. При этом последствия технического долга могут распространяться за пределы отдельных программных компонентов, в том числе затрагивая связи с архитектурой предприятия.

Современные подходы к управлению техническим долгом ИТ-систем включат статистический анализ кода, выявление архитектурных нарушений и зависимостей, количественную оценку существующих ограничений, мониторинг технического состояния компонентов ИТ-систем, приоритезацию мероприятий по модернизации и применение методов машинного обучения для прогноза динамики технического долга [14, 17]. Вместе с тем развитие архитектуры предприятия требует учитывать возможную динамику накопления технического долга при разных изменениях ИТ-ландшафта, включая также изменения связей между компонентами, «устаревание» технологий, вопросы встраивания программных систем в структуру бизнес-процессов предприятия.

Перспективным направлением решения этой задачи является применение методов машинного обучения, в том числе нелинейных. В частности, метод случайного леса (Random Forest) позволяет формировать прогнозы на основе совокупности признаков разного рода, в том числе характеризующих архитектурные особенности и возможности эксплуатации информационных систем. Использование этого подхода может создать действия по переходу от реактивного устранения технического долга к прогнозно-ориентированному управлению, который обеспечивает выборность его накопления. Интеграция методов машинного обучения с инструментами анализа архитектуры предприятия представляет особый интерес с точки зрения повышения обоснованности решений о модернизации информационных систем и распределении ограниченных ресурсов предприятия.

Целью исследования является разработка методики прогнозирования технического долга информационных систем на основе анализа характеристик архитектуры предприятия и применения методов машинного

обучения. Предлагаемая методика ориентирована на выявление ИТ-систем с потенциально высоким уровнем технического долга, оценку перспектив его изменения и информационную поддержку принятия решений при планировании архитектурных преобразований.

II. Материалы и методы исследования

Основу данного исследования научные публикации, посвящённые архитектуре предприятия, управлению информационными системами, техническому долгу ИТ-систем и применению методов машинного обучения. В исследованиях H. Jonkers и соавторов [12], а также M. Lankhorst и соавторов [13] архитектура предприятия рассматривается как совокупность взаимосвязанных представлений бизнес-процессов, приложений, данных и технологической инфраструктуры. Авторы указывают на необходимость согласования архитектурных доменов, поскольку изменения отдельных компонентов в рамках информационных систем способны затрагивать смежные элементы предприятия. Указанные положения создают теоретическую основу для рассмотрения технического долга на уровне программного кода как межсистемной зависимости архитектурных ограничений.

Вопросы результативности управления архитектурой представлены в исследованиях C. Schmidt и P. Buxmann [15], T. Tamm и соавторов [16]. На основе эмпирических данных, организации показывают значимость работы по управлению архитектурой для обеспечения гибкости и эффективности корпоративной среды. Исследователи предлагают модель создания организационной ценности архитектуры предприятия, выделяя согласованность всех организационных элементов, доступность информации, оптимизацию портфеля ресурсов и их пополняемость.

Большое значение для данного исследования имеют работы S. Kotusev [10, 11], S. Kurnia [11] и Y. Gong, M. Janssen [9], посвящённые методологическим основаниям архитектурной практики. Kotusev в рамках исследования выявил расхождения между формализованными частями методологий архитектуры и особенностями применения их в практической деятельности. Совместно с Kurnia, он систематизировал теоретические подходы, объясняющие использование архитектурных артефактов и их роль в принятии решений. Gong и Janssen обратили внимание на слабую эмпирическую обоснованность части положений об эффективности архитектуры предприятий и важность исследования механизмов формирования их ценности.

В российской научной литературе вопросы развития архитектуры предприятия представлены в работах Д.В. Кудрявцева и М.Ю. Арзуманяна [4], А.Б. Анисифорова [1], Л.З. Давлеткиреевой и соавторов [2]. Исследователи рассматривают эволюцию архитектурного подхода от проектирования ИТ-инфраструктуры к осуществлению поддержки комплексных преобразований, в том числе через планирование архитектуры предприятия, принцип эксплуатации архитектуры, подчёркивая необходимость согласования потребностей организации и возможностей её технологической среды. Несмотря на это, эти работы не предлагают специализированной модели по прогнозу технического долга архитектуры предприятия.

Для реализации задач данного исследования важно рассмотреть научные работы, связанные с накоплением архитектурного технического долга. В работе A. Martini, J. Bosch и M. Chaudron [14] на основе эмпирического наблюдения за работой пяти крупных компаний были рассмотрены факторы возникновения и накопления архитектурного долга, а также особенности устранения этого долга в процессе развития программных систем. Основной акцент авторы уделяют проблемам выделения приоритетов в краткосрочных функциональных проектах относительно долгосрочного обеспечения технического качества архитектуры и предлагают модели накопления и сокращения технического долга. По мнению авторов, процесс устранения архитектурных ограничений нередко имеет реактивный характер и осуществляется после того, как возникли существенные затруднения в разработке.

Перспективы применения машинного обучения к проблеме технического долга представлены в научном исследовании В.В. Качанова и др. [3], в котором авторы рассматривают модели выявления и исправления разных дефектов программного обеспечения, которые связаны с накоплением технического долга. Несмотря на это, этот подход преимущественно ориентирован на особенности исходного кода и не охватывает полноценно взаимосвязи между бизнес- и технологическим уровнями.

Научная работа D. Tsoukalas, D. Kehagias, M. Siavvas и A. Chatzigeorgiou [17] посвящена прогнозированию динамики технического долга – на основе еженедельных наблюдений за пятнадцатью проектами. В результате, было подтверждено, что линейные регулярные модели способны обеспечивать содержательные прогнозы при относительно коротких горизонтах, тогда как регрессия на основе случайного леса демонстрировала лучшие данные по сравнению с иными линейными моделями при более продолжительном прогнозировании. Предпосылки применения алгоритма случайного леса представлены в работе L. Breiman [8], в которой обоснован ансамблевый подход к созданию моделей классификации и регрессии на основе деревьев решений, представляет интерес в теме прогнозирования технического долга ИТ-систем.

Проведённый анализ научных публикаций позволяет выделить исследовательскую задачу, заключающуюся в разработке методики, объединяющей количественную оценку технического долга, характеристики архитектуры предприятия и инструменты машинного обучения. В отличие от подходов, ориентированных на исходный код, предлагаемая постановка предусматривает учёт архитектурных ограничений, зависимостей систем и параметров технологического развития.

Методами исследования составили архитектурный и системный подходы, методы сравнительного анализа, статистики и машинного обучения. Для количественной оценки технического долга программных систем были применены характеристики архитектуры предприятия, в том числе связанность компонентов, сложность разных интеграций, технологическую устарелость, покрытие ИТ-ландшафта автоматизированным тестированием и частоту изменений архитектуры. Прогноз технического долга производился методом случайного леса (Random Forest) на основе исторических данных об архитектуре и величине накопленного технического долга. Для оценки качества прогнозирования предусматривался расчёт по спектру показателей (*MAE, RMSE,* R^2) – для определения точности модели и основания приоритетов изменений.

III. Методика прогнозирования технического долга

Методика по прогнозированию технического долга информационных систем основывается на комплексном рассмотрении технических и организационных факторов архитектуры предприятия в разрезе с экономическими характеристиками. В техническом аспекте исследуются состояние программных компонентов, интеграционные взаимодействия, используемые технологии, ограничения архитектуры, возникающие при эксплуатации различных информационных систем. С позиции организационного аспекта рассматривается влияние накопленного долга на реализацию бизнес-процессов, возможности развития и модернизации информационных систем и потребности в дополнительных ресурсах. Указанные изменения совсем не подменяют друг друга: основным объектом прогноза является архитектурно-технический долг, трудозатраты и бизнес-критичность используются при интерпретации прогноза и принятия управленческих решений. Синтез этих направлений позволяет рассматривать технический долг как особенности несовершенства программных решений и как фактор, влияющий на эффективность архитектурного управления предприятием.

Предлагаемая методика предполагает организацию постепенного перехода от анализа существующего ИТ-ландшафта к количественной оценке технического долга и прогнозированию его дальнейшего изменения. Метод охватывает архитектуру информационных систем, в том числе с техническими ограничениями. Специалисты по информационным технологиям изучают, как копятся в этих системах ошибки, и как они мешают организации работать. Благодаря полученным результатам эксперты сопоставляют разные пути модернизации и выбирают те решения, которые лучше всего исправят накопленные недостатки и ускорят работу.

Развитие архитектурного ландшафта происходит на основе фактов. Сначала архитекторы описывают состав инфраструктуры предприятия. Сотрудники составляют перечень программных модулей и определяют, как они связаны друг с другом. Далее они должны выяснить, за какие процессы отвечает каждый сервис. Это помогает понять, насколько важны технические узлы для бизнеса. Эксперты изучают текущую схему работы, ищут старые программы или функции, которые дублируются. Лишние связи формируют в том числе технический долг. Позже на основе полученных данных формируется целевая модель. В специальный реестр вносят технические запреты, такие, как: неподдерживаемые средства и API-контракты с нарушениями, фиксируются компоненты-двойники или случаи, когда системы слишком сильно зависят от внешних условий. Старую программу или множество связей не называют долгом автоматически. Проблема появляется только тогда, когда код мешает развитию или требует срочного исправления.

Далее необходимо внедрение системы архитектурных метрик. На основе факторов (табл. 1) она позволяет измерить качество комплекса программных средств и понять, как возможно построить целевую и транзитную архитектуру. Каждый показатель рассматривается как потенциальный фактор прогноза и должен иметь способ измерения. Показатели рассчитаны на регулярное наблюдение в единицах измерения, сопоставимых друг с другом, их состав может быть уточнён с учётом данных по особенностям ИТ-ландшафта. В постановке прогноза все факторы выступают как объясняющие признаки, а не автоматически суммируемые составляющие целевого технического долга.

ТАБЛИЦА I. Характеристики архитектуры предприятия, используемые для прогнозирования технического долга

Группа факторов	Показатель	Способ измерения
Жизненный цикл	«Возраст» информационной системы	Количество лет с момента ввода ИТ-системы в эксплуатацию
Архитектурная сложность	Связанность компонентов	Число связей между компонентами относительно их общего числа
Интеграционная сложность	Количество интеграций	Число интерфейсов и интеграционных взаимодействий
Технологическая устарелость	Доля устаревших технологий	Доля компонентов, использующие технологии, требующие замены или не имеющие интеллектуальной поддержки
Сопровождаемость	Доля использования автоматизированного тестирования	Доля программного кода или компонентов, охваченных автоматизированным тестированием
Изменчивость архитектуры	Частота изменений	Количество архитектурных изменений за период
Надёжность	Плотность дефектов	Количество зарегистрированных дефектов на выбранную единицу объёма системы

Примечание: составлено авторами

Для оценки технического долга информационных систем на количественном уровне авторами предлагается использовать подход, предусматривающий выделение следующих составляющих: технических, ресурсных и бизнес-ориентированных. Техническая составляющая в данном контексте характеризует накопленные нарушения архитектуры с учётом их критичности масштаба влияния на информационные системы, ресурсная определяется по параметрам трудоёмкости устранения соответствующих ограничений. Бизнес-ориентированная составляет в итоге предполагает учёт критичности программных систем для реализации бизнес-процессов предприятия и применяется при определении приоритетов архитектурных изменений. Разграничение данных составляющих позволяет в общем-то комплексно оценивать состояние ИТ-систем, делать прогноз технического долга и обосновывать решения для модернизации ИТ-ландшафта предприятия.

Непосредственную техническую оценку предлагается рассчитывать по реестру ограничений, которые не были устранены на системном уровне. Каждому ограничению j в информационной системе i на момент t присваиваются балл критичности s и балл масштаба влияния a, а потом определяется суммарный индекс технического долга. При этом, значение критичности отражает серьёзные ошибки и нарушения в архитектурных требованиях, а масштаб их влияния – число или значимость затронутых компонентов и

интеграций. Оба параметра могут оцениваться в рамках единой пятибалльной шкалы. Правила присвоения баллов и перечень учитываемых ограничений (*m*) фиксируются до выполнения соответствующих расчётов:

$$TD_{i,t}^{tech} = \sum_{j=1}^{m_{i,t}} s_{j,t}\ a_{j,t}.$$

Индекс архитектурно-технического долга показывает степень накопления дефектов в рамках информационной системы, но не оценивает в денежном эквиваленте спектр потерянных возможностей и необходимое число штатных специалистов, которых нужно будет привлечь для работы над устранением дефектов. Для сравнения систем разной размерности и их принадлежности в работе предприятия допускается нормировать этот индекс за счёт плотности долга – отношения индекса долга к числу компонентов архитектуры в той же системе при постоянных границах. Рассматриваемые ограничения в рамках анализируемых информационных систем могут служить причинами для накопления последующего технического долга, однако для этого будут нужны новые расчёты на основе данных, полученных в новых случаях мониторинга.

Параллельно с индексом архитектурно-технического долга необходимо вычислить ресурсный показатель. Для этого специалисты должны оценить объём привлекаемых усилий, чтобы полностью убрать мешающие барьеры и очистить архитектуру от накопившихся проблем. Данный показатель измеряют в человеко-днях. Сюда включается: обновление технологий и исправление дефектов, также инженерам предстоит вручную согласовывать экраны и интерфейсы, переносить массивы данных и проверять, как работает система с имеющимся программным кодом.

Данная метрика помогает планировать бюджет для модернизации архитектурного ландшафта. С помощью этой цифры руководители понимают, сколько людей и средств нужно привлечь, чтобы вовремя и качественно улучшить рабочую среду. Оценивать технический долг исключительно только трудозатратами (*e*) нельзя, так как одинаковые трудозатраты могут соответствовать разным уровням критичности и масштабам влияния на систему. Часто одинаковое время тратят на исправление мелких багов и системных сбоев, которые неодинаково влияют на то, выдержит ли продукт нагрузку.

$$TD_{i,t}^{res} = \sum_{j=1}^{m_{i,t}} e_{j,t}\ .$$

Бизнес-ориентированное измерение фокусируется на выборе приоритетов преобразований. Каждой системе даётся коэффициент бизнес-критичности *B*, отражающий роль в обеспечении значимых процессов, непрерывности операций и работе конкретных функций предприятия. Произведение прогнозного технического индекса на коэффициент *B* образует вспомогательный показатель приоритетности вмешательства, он позволяет отличать критичные для бизнеса системы от вспомогательных при одинаковом уровне архитектурных нарушений. Этот показатель не выступает самостоятельной величиной долга и при принятии инвестиционных решений должен дополняться стоимостью изменений и оценкой рисков:

$$P_{i,t} = \widehat{TD}_{i,t+h}^{tech}\ B_i,$$

где: $P_{i,t+h}$ – прогнозируемый показатель приоритета действий по архитектурному вмешательству, B_i – коэффициент бизнес-критичности, *i* – номер ИТ-системы предприятия, *t* – период наблюдения, *h* – горизонт прогнозирования, а $\widehat{TD}_{i,t+h}^{tech}$ – прогнозная величина технического долга ИТ-системы в архитектуре.

Для прогнозной модели важно создать вектор разных признаков архитектуры $X_{i,t}$, включающий показатели из таблицы 1 – по каждой информационной системе, а также периоду наблюдения. В частности, необходимо включить в этот спектр показателей возраст системы, связанность её компонентов, число имеющихся интеграций, процент устаревших технологий, покрытие автоматизированными тестами, частота архитектурных изменений и плотность дефектов. Показатели с разными единицами измерения сохраняются в отдельных компонентах вектора – правила их расчёта должны быть неизменными для обучающей и проверочной выборок. При нужности в модель могут быть включены дополнительные характеристики системы – такие, как масштаб, исторические значения индекса долга, доступные на момент прогнозирования:

$$\mathbf{X}_{i,t} = \left(x_{i,t,1};\ x_{i,t,2};\ \ldots;\ x_{i,t,p}\right)$$

Данные для модели можно собирать из архитектурных репозиториев, реестров технических ограничений, систем по управлению изменениями и инцидентами, средств контроля качества кода и документации по интеграциям. Эти записи сопоставляются по личному идентификатору информационной системы и периоду времени, а ограничения, затрагивающие несколько компонентов, учитываются по установленному правилу, без повтора подсчёта одного и того же долга. Подготовка выборки включает в себя: проверку пропусков, согласование периодичности наблюдений и контроль изменений в границах систем. В случае использования синтетических данных расчёты выступают демонстрационными, без переноса полученной точности модели на реальные ИТ-ландшафты организации.

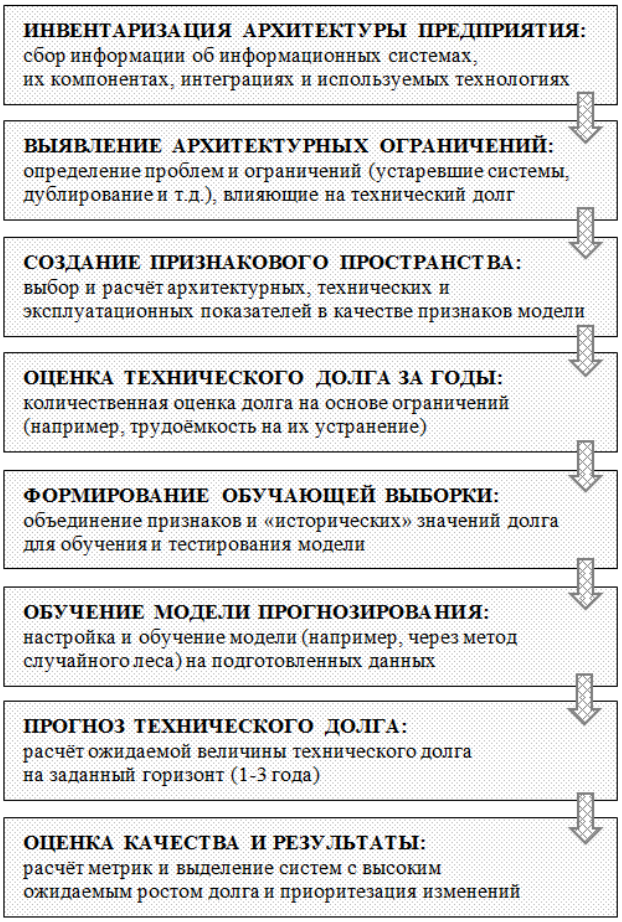
Обучающая выборка строится из пар, где значения архитектурных признаков на момент *t* сопоставлены с фактическим индексом технического долга этой же информационной системы за период *t+h*. Горизонт *h* определяется в соответствии с задачами планирования и достатком архивных наблюдений, а вот для многолетнего прогноза требуются данные, которые охватывают конкретную динамику. Обучающая и проверочная части разделяются по времени, чтобы сведения из будущих периодов не попадали в модель в качестве входных признаков. При наличии нескольких систем необходимо также учитывать зависимость наблюдений, относящихся к одному ИТ-ландшафту:

$$D = \left\{\left(\mathbf{X}_{i,t}, TD_{i,t+h}^{tech}\right)\right\}_{(i,t)\in I},$$

где: *D* – обучающая выборка, сформированная на основе архитектурных характеристик ИТ-систем и значений

технического долга, а I – множество допустимых пар «программная система – период наблюдения».

Подытоживая, последовательность действий показана на рисунке: после инвентаризации архитектуры выявлены ограничения, формируется признаковое пространство и рассчитываются «исторические» значения долга, затем подготавливается выборка, обучается модель, строится прогноз и оцениваются результаты. При количественной оценке целесообразно фиксировать два связанных, но и не связанных друг с другом результата – технический индекс в условных баллах и трудоёмкость устранения различных ограничений в человеко-днях. После построения системы прогноза технического индекса осуществляется бизнес-интерпретация с учётом критичности поддерживаемых процессов и возможностей финансирования изменений.



Примечание: составлено авторами

Рис. 1. Методика прогнозирования технического долга информационных системна основе характеристик архитектуры предприятия

В качестве ключевого метода прогнозирования был выбран Random Forest – ансамбль, включающий в себя набор регрессионных деревьев решений. Каждое дерево строит прогноз будущего индекса технического долга по имеющемуся вектору признаков, а итоговый результат определяется усреднением результатов деревьев.

Данный алгоритм позволяет учитывать нелинейные зависимости и взаимодействия архитектурных данных, вместе с тем его преимущество не может считаться как заранее установленное и требует обязательного анализа и сопоставления с базовыми методами прогнозов, иными регрессионными моделями. Расчёты идут по формуле:

$$\widehat{TD}^{tech}_{i,\,t+h} = \frac{1}{K}\sum_{k=1}^{K} T_k\left(\mathbf{X}_{i,t}\right),$$

где, в дополнение: K – общее число деревьев решений в ансамбле случайного леса, k – порядковый номер дерева решений, принимающий значения от 1 до K, и также T_k – прогноз технического долга, полученный с помощью k-го дерева решений.

Оценка качества строится на сравнении рассчитанных прогнозов с текущими значениями технического индекса в отложенных периодах. В итоге, средняя абсолютная ошибка *MAE* характеризует типичное отклонение этого прогнозного значения от наблюдаемого, *RMSE* – в том числе усиливает вклад крупных ошибок, а коэффициент детерминации R^2 позволяет оценить модель в отношении прогноза по среднему значению проверочной выборки. Показатели *MAE* и *RMSE* выражаются в баллах долга, а R^2 безразмерен и на проверочных данных может иметь отрицательные значения. В целях прикладного характера следует также анализировать ошибки по программным системам и сравнивать модель с простым прогнозом по сохранению последнего наблюдаемого уровня долга. В дополнение к вышеуказанным условным обозначениям, в рассматриваемых формулах имеются q – это порядковый номер наблюдения в проверочной выборке, принимает в том числе значения от 1 до n, и n – общее количество наблюдений в проверочной выборке, в основе которых оценивается качество прогнозирования.

Формула средней абсолютной ошибки:

$$\mathrm{MAE} = \frac{1}{n}\sum_{q=1}^{n}\left|TD_q - \widehat{TD}_q\right|.$$

Формула среднеквадратичной ошибки:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{q=1}^{n}\left(TD_q - \widehat{TD}_q\right)^2},$$

Коэффициент детерминации:

$$R^2 = 1 - \frac{\sum_{q=1}^{n}\left(TD_q - \widehat{TD}_q\right)^2}{\sum_{q=1}^{n}\left(TD_q - \overline{TD}\right)^2}.$$

Условные обозначения: TD_i – фактическое значение технического долга для i-го наблюдения, $\overline{TD}$ – среднее фактическое значение технического долга в проверочной выборке, а $\widehat{TD}_i$ – прогнозируемое значение долга.

Прогнозируемое изменение индекса позволяет в итоге выявлять системы, для которых вероятно сохранение или усиление архитектурных ограничений. Однако оценка у будущего долга не должна автоматически трактоваться как эффект конкретного мероприятия по модернизации: для такого вывода необходимы данные о проведённых изменениях и отдельное сценарное сравнение. Ресурсная оценка в человеко-днях и бизнес-критичность могут быть

использованы после прогнозирования для анализа в том числе целесообразности вмешательства, для определения последовательности работ и согласования инициатив по работе с техническим долгом – между ИТ-специалистами и владельцами бизнес-процессов. Таким образом, данная диагностика будет дополняться экономическим видением без смешения различных единиц измерения.

Основные этапы практического применения методики и ожидаемые результаты их выполнения сведены в табл. 2. Предлагаемая последовательность формирует единый контур: регистрация архитектурных ограничений, расчёт технического и ресурсного показателей, построение «исторической» выборки, прогнозирование будущего технического индекса, проверка качества и выбор направлений модернизации. На техническом уровне результатом становится прогноз накопления архитектурных нарушений, на ресурсном – оценка потребности в работах по их устранению, а на уровне бизнес-архитектуры – обоснование приоритетов архитектурных инициатив. Полноценная эмпирическая проверка эффективности контура требует сопоставимых «исторических» данных, поэтому демонстрационные расчёты на синтетических наблюдениях рассматриваются исключительно как иллюстрация его вычислительной реализации.

ТАБЛИЦА II. ЭТАПЫ ПРИМЕНЕНИЯ МОДЕЛИ ПРОГНОЗИРОВАНИЯ ТЕХНИЧЕСКОГО ДОЛГА

Содержание	**Результат**
Подготовка данных	
Сбор характеристик архитектуры и истории изменений	Набор исходных наблюдений
Оценка целевой переменной	
Расчёт трудоёмкости устранения архитектурных ограничений	«Исторические» значения технического долга
Подготовка выборки	
Составление признаков с будущими значениями технического долга	Обучающая и тестовая выборки
Обучение модели	
Настройка метода случайного леса и обучающих данных	Обученная модель
Проверка прогноза	
Расчёты *MAE, RMSE,* R^2 на отложенных данных	Оценка качества реализованной модели
Использование результата	
Выделение систем с ожидаемым ростом технического долга	Приоритеты архитектурных изменений

Примечание: составлено авторами.

Используя эту модель, можно находить программные системы, в которых технические рамки быстро сужаются и модули связываются слишком тесно. Технологии в них довольно быстро стареют. Если эксперты оценивают весь объём накопившегося долга заранее, они наглядно видят разницу между нынешним и будущим ИТ-ландшафтом, что помогает вовремя обновлять код и не давать разным преградам возникнуть вновь. Сравнивать архитектурные показатели помогают количественные методы принятия решений, включая расчёт оценки последствий.

Эта методика связывает качество кода с финансовыми затратами и тем, как реализуются бизнес-процессы. Тут специалисты определяют, сколько времени они потратят на исправление архитектуры. Дополнительная оценка помогает точнее распределять ресурсы. Если понимать, какую роль играет конкретная программа в организации, можно определить вес каждой технической правки. При этом, анализируя технические, денежные и показатели бизнеса, команда расставляет верные приоритеты. Как и в каком порядке внедрять изменения, решают ведущие специалисты, стараясь максимально согласовать новые задачи с глобальной стратегией всей компании. Прогноз технического долга стало рабочим инструментов для управления всей архитектурной системой предприятия.

Практическое применение методики предполагает её интеграцию в процессы архитектурного управления предприятием, включая деятельность архитектурного комитета, формирование портфеля инициатив и планирование перехода от текущего состояния архитектуры к целевому. Результаты прогнозирования могут использоваться при сравнении альтернативных сценариев модернизации, определении информационных систем, требующих первоочередного архитектурного вмешательства, и оценке последствий откладывания необходимых технических изменений. Периодическое обновление архитектурных показателей и повторное построение прогнозов позволяют организовать анализ и мониторинг образования и накопления технического долга, изменять направления развития ИТ-ландшафта.

IV. РЕЗУЛЬТАТЫ АПРОБАЦИИ И ОБСУЖДЕНИЕ

Авторам удалось оценить методику на практике [7]. Они изучили архитектурный ландшафта компании «СИДЖИО Консалтинг» («CGO Consulting» LLC), чтобы понять специфику работы её цифровых систем. В рамках анализа требовалось найти уязвимые места. В связи с этим эксперты анализировали текущие бизнес-процессы и беседовали с персоналом. Специалисты также смогли подготовить архитектуры схемы на языке ArchiMate и провели GAP-анализ, благодаря чему удалось вовремя выявить технологические и интеграционные барьеры, из-за которых обычно возникает технический долг.

Была проведена работа по трём проектам. В рамках первого проекта предусматривалась замена устаревшей информационной системы управления проектами новым программным решением. В ходе исследования выявлены технологические зависимости, сложности миграции всех «исторических» данных и необходимость обеспечения их совместимости с корпоративными сервисами. Ещё один проект был связан с переходом корпоративной системы от монолитной к микросервисной архитектуре. В рамках анализа были выделены основные проблемы: избыток связей между компонентами, сложность в обеспечении функционального расширения и ограниченные способы интеграции с другими информационными системами. И, наконец, третий проект – внедрение гибридной модели интеграции корпоративных информационных систем. В процессе анализа были выявлены дублирование данных, несогласованность правил информационного обмена и отсутствие общего механизма управления интеграциями.

В соответствии с разработанной методикой вначале была количественная оценка технической составляющей долга на основе выявленных ограничений архитектуры. Для каждого нарушения определяются критичность и масштаб влияния на информационную систему, и эти показатели принимают значения от 1 до 5 баллов. При этом критичность характеризует серьёзность нарушения

архитектурных требований, тогда как масштаб влияния отражает распространение последствий на программные компоненты, интеграции и функциональные компоненты информационной системы. В общем итоге формируется интегральный показатель архитектурно-технического долга, рассчитываемый посредством суммирования всех произведений указанных характеристик.

Рассмотрим вопрос реализации данной методики на практическом примере с использованием синтетических данных, сформированных с учётом характера различных ограничений, выявленных в рассмотренных проектах.

На первом проекте условные оценки по критичности и масштабу влияния трёх ограничений составляют (4; 3), (3; 2) и (4; 2) соответственно. Для второго проекта будут приниматься пары значений (5; 4), (4; 4) и (4; 3), в третьем проекте – (4; 2), (3;2) и (3; 3). В соответствии с формулой оценки технической составляющей получаем значения:

$$TD_{1,\,t}^{tech} = 4 \times 3 + 3 \times 2 + 4 \times 2 = 26 \text{ баллов},$$

$$TD_{2,\,t}^{tech} = 5 \times 4 + 4 \times 4 + 4 \times 3 = 48 \text{ баллов},$$

$$TD_{3,\,t}^{tech} = 4 \times 2 + 3 \times 2 + 3 \times 3 = 23 \text{ балла}.$$

Эти данные характеризуют выраженные нарушения в архитектуре с учётом их критичности и воздействия. При этом наибольшая условная величина технического долга соответствует второму проекту – 48 баллов, что связано с рассматриваемыми проблемами избыточной связности компонентов и сложности модернизации монолитной информационной системы. По результатам расчётов 26 баллов у первого проекта, а у третьего – 23 балла. Таким образом, предложенный показатель позволяет сравнить архитектурные ограничения разных ИТ-систем по числу и степени потенциального воздействия на техническое состояние ИТ-ландшафта предприятия.

Следующим этапом осуществляется оценка ресурсов для устранения технического долга, определяемой через трудоёмкость устранения выявленных ограничений. Для сохранения сопоставимости с проведёнными расчётами примем условные значения трудозатрат по ограничениям в количестве трёх позиций: для первого проекта – 12, 8 и 5 человеко-дней, для второго – 16, 10 и 8 человеко-дней, для третьего – 9, 6 и 3 человеко-дня. По итогу получаем следующие значения:

$$TD_{1,\,t}^{res} = 12 + 8 + 5 = 25 \text{ чел./дн.},$$

$$TD_{2,\,t}^{res} = 16 + 10 + 8 = 34 \text{ чел./дн.},$$

$$TD_{3,\,t}^{res} = 9 + 6 + 3 = 18 \text{ чел./дн.}$$

Сравнение технических и ресурсных оценок может показать, что указанные показатели характеризуют в том числе разные аспекты накопленного технического долга. Интегральный технический индекс отражает масштаб архитектурных ограничений и их критичность, тогда как ресурсная составляющая характеризует предполагаемую трудоёмкость их устранения. Например, второй проект имеет наибольшие значения показателей, однако такое совпадение не является обязательным: информационная система может обладать сравнительно высоким уровнем архитектурных нарушений при относительно небольшой трудоёмкости их устранения либо, напротив, требовать значительных ресурсов для исправления ограниченного количества технических проблем.

Для построения прогнозной модели каждому из всех наблюдений ставится в соответствие вектор признаков архитектуры, включающий семь показателей, которые представлены в табл. 1. В качестве иллюстрации – условный вектор первого проекта принимает значения:

$$\mathbf{X}_{1,\,t} = (6;\ 0{,}35;\ 12;\ 0{,}30;\ 0{,}65;\ 6;\ 2).$$

Векторы для других проектов по ИТ-системам будут формироваться аналогично. Обучающая выборка имеет сопоставление указанных признаков с последующими значениями технического индекса, определяемого через зарегистрированные архитектурные нарушения.

Покажем работу математической модели на примере. С помощью алгоритма случайного леса мы предскажем, как вырастет долг ИТ-системы в течение ближайшего года. Если три регрессионных дерева выдадут для первого объекта оценки 27, 29 и 31, для второго – 47, 50 и 53, а для третьего – 24, 27 и 30 баллов, итоговое число мы получим, когда вычислим среднее арифметическое этих значений:

$$TD_{1,\,t+h}^{tech} = \frac{27 + 29 + 31}{3} = 29 \text{ баллов},$$

$$TD_{2,\,t+h}^{tech} = \frac{47 + 50 + 53}{3} = 50 \text{ баллов},$$

$$TD_{3,\,t+h}^{tech} = \frac{24 + 27 + 30}{3} = 27 \text{ баллов}.$$

Вычислим, как изменится архитектурный долг за выбранный срок. У первой программы разница составит 3 балла, вторая прибавит 2, а третья увеличится на 4. Хотя сумма накопленных ограничений у третьего объекта изначально ниже, его показатели растут быстрее всего. В таких случаях специалисты изучают текущее состояние программных систем и заранее прогнозируют будущие риски.

Руководители обосновывают шаги, когда соотносят технические метрики со значимостью системы для компании. Оценим проекты по пятибалльной шкале. Если мы присвоим первому объекту коэффициент 4, второму – 3, а третьему – 5, мы сможем рассчитать, насколько срочно нужно менять код или структуру каждой программы.

$$P_{1,\,t+h} = 29 \times 4 = 116,$$

$$P_{2,\,t+h} = 50 \times 3 = 150,$$

$$P_{3,\,t+h} = 27 \times 5 = 135.$$

Математика определяет очередность работ. Вторая система лидирует в списке, за ней следует третья, а закрывает перечень первая. Высокая важность процессов внутри третьей программы делает ее приоритетнее первой, даже если сам технический долг там меньше. Когда ИТ-инструменты привязаны к прибыли или производству, организация точнее понимает цифры и эффективнее управляет технологиями.

Для принятия управленческих решений полученные показатели необходимо дополнительно сравнить в том числе с ресурсами, архитектурными зависимостями, а также принять во внимание риски при реализации новых

преобразований. В данном примере второй проект имеет наибольший показатель приоритетности и одновременно требует наибольшего объёма ресурсов для устранения выявленных ограничений – 34 человеко-дня. При этом третий проект обладает меньшей ресурсной оценкой – 18 человеко-дней – при достаточно высоком значении по бизнес-приоритетности. Следует, что при ограниченных ресурсах организация может рассматривать поэтапную реализацию преобразований в рамках второй проекта с параллельным устранением наиболее критичных задач третьего проекта. Окончательный выбор должен будет учитывать бюджет, сроки, технические зависимости и допустимые риски изменения архитектуры.

Содержательная интерпретация данных результатов позволяет определить возможные направления по росту каждого из проектов. Для первого проекта приоритетные мероприятия могут быть ориентированы на подготовку миграции данных, устранение технологических связей, проверка совместимости интеграционных интерфейсов и обеспечение непрерывности связанных процессов. Второй проект стоит развивать поэтапно. Архитектуру разделяют на части, а функциональные модули делают полностью автономными, чтобы потом разработчики могли снизить зависимость элементов и внедрить единый стандарт для API. В третьем проекте необходимо унифицировать связи и удалить лишние данные. Также специалисты описывают правила обмена информацией и следят за качеством потоков. Подобные меры помогают бороться с техническим долгом. Команды внедряют их, когда проанализируют затраты и изучат последствия.

Чтобы оценить качество прогнозной модели, возьмем три примера. Когда реальные значения долга равны 34, 55 и 29 баллам, а прогноз ранее составил 29, 50 и 27, математический расчет выявит среднюю абсолютную ошибку:

$$\mathrm{MAE} = \frac{5 + 5 + 2}{3} = 4 \text{ балла},$$

а корень средней квадратичной ошибки равен

$$\mathrm{RMSE} = \sqrt{\frac{25 + 25 + 4}{3}} \approx 4{,}24 \text{ балла}.$$

Среднее контрольное значение технического долга составляет 39,33 балла, а коэффициент детерминации $R^2 \approx$ 0,858, хотя эти параметры были математически установлены через синтетические данные. Такие данные используются лишь для того, чтобы показать на практике применение сформулированной методики.

Численные расчеты помогают сравнить три инициативы. Так мы отделяем ошибки в структуре кода от финансовых трат и общей значимости систем для организации. Методика позволяет измерять накопленный груз. Одновременно с этим она помогает предсказывать его объем в будущем. Когда аналитики соотносят технологический индекс с доступными ресурсами, они определяют приоритетную систему. Очередность работ также зависит от того, насколько критична конкретная программа для бизнеса.

Описанные вычисления имеют свои границы. Мы использовали искусственные цифры, так как у нас отсутствовала полноценная история наблюдений за тем, как меняются программные комплексы. Синтетика лишь демонстрирует рабочий алгоритм, который в будущем пройдет проверку. Чтобы подтвердить методику, потребуется создать общий реестр архитектурных нарушений и внедрить единую систему их оценки. На основе собранных сведений эксперты подготовят выборки и обучат модель «случайного леса». Точность такого прогноза специалисты сравнят с показателями других систем на контрольных группах данных.

Таким образом, результаты аналитической апробации позволяют обосновать применимость данной системы архитектурных показателей к описанию технических и иных ограничений информационных систем и показать возможность использования результатов их оценки в процессах архитектурного управления. Интеграция трёх составляющих (технической, бизнес-ориентированной и ресурсной) создаёт условия для создания прогнозного механизма поддержки принятия решений, где в рамках него будет осуществляться модернизация программных систем с учётом накопленных ограничений, ожидаемой динамики расширения технического долга и значимости отдельных компонентов для деятельности предприятия.

V. Заключение

В результате проведённого исследования разработана методика количественной оценки и прогнозирования технического долга информационных систем с учётом архитектурных характеристик. Авторской особенностью предложенного подхода выступает комплексный анализ технических, ресурсных и бизнес-составляющих, так как они обеспечивают переход от оценки трудоёмкости по устранению ограничений к анализу критичности, а также оценке масштаба воздействия и значимости в конкретной организации. Кроме того, была сформирована система архитектурных показателей, позволяющая формировать характеристику состояния программных систем, делать анализ и выявлять потенциальные факторы накопления технического долга и создавать основу для построения прогнозных моделей. Для обеспечения прогнозирования предложено применение метода случайного леса.

Методика была проверена на примере трёх проектов, данные по которым были обезличены. Авторы смогли измерить при помощи синтетических данных, как может архитектура ограничивать работу системы и ландшафта, установили нужный для измерения объём ресурсов. На основе полученной информации спрогнозировали долг и определили, каким проектам нужно выделить внимание в первую очередь. Итоги показали: технические данные программ часто не соответствуют их реальной роли для бизнеса. Чтобы подтвердить работоспособность модели, необходимо изучить накопленные за прошедшие годы сведения и детально сравнить результаты со способами предсказывать качество программного кода.

Благодаря данной методике специалисты компании могут спланировать изменения в ИТ-ландшафте, выбрав приоритетные направления для устранения технического долга и грамотно распределив трудовые ресурсы. Подход по реализации изменений меняется – специалисты теперь перестают устранять лишь дефекты, начиная управлять техническим долгом на опережение, ориентируясь также на бизнес-приоритеты и цели организации.

В дальнейшем авторы планируют испытать данные алгоритмы на реальных кейсах. Предстоит выяснить, как конкретные факторы влияют на прогноз, чтобы встроить методику в математический аппарат, который помогает архитектурному комитету и отдельным специалистам в том числе быстрее выбирать верные действия.

Список литературы

[1] Анисифоров А.Б. Базовые принципы формирования, развития и эксплуатации информационной инфраструктуры предприятия в решении задач информационного менеджмента // Научный журнал НИУ ИТМО. Серия: Экономика и экологический менеджмент. 2019. № 3(38). С. 128–136.

[2] Давлеткиреева Л.З., Новикова Т.Б., Курзаева Л.В., Лактионова Ю. С., Подкользина Л. В. Управление технологической архитектурой организации на основе усовершенствованного метода Enterprise Architecture Planning // Фундаментальные исследования. 2015. № 8-2. С. 252–256.

[3] Качанов В.В., Марков С.И., Цурков В.И. Машинное обучение в проблеме технического долга программного обеспечения // Известия РАН. Теория и системы управления. 2023. № 4. С. 98–104. 10.31857/S0002338823040078

[4] Кудрявцев Д.В., Арзуманян М.Ю. Архитектура предприятия: от проектирования ИТ-инфраструктуры к трансформации бизнеса // Российский журнал менеджмента. 2017. № 15(2). С. 193–224. 10.21638/11701/spbi18.2017.204

[5] Мамедгулиев Р.И., Сажнева Л.П. Использование инструментов машинного обучения для реализации бизнес-процессов компании в контексте архитектуры предприятия // Мягкие измерения и вычисления. 2025. № 92(7). С. 69–86. 10.36871/26189976.2025.07.007

[6] Мамедгулиев Р.И., Цуканова О.А. Исследование и анализ взглядов ИТ-специалистов на системную, корпоративную и бизнес-архитектуру предприятия: тенденции, проблемы, перспективы развития // Бизнес-информатика. 2026. № 20(3). С. 69–89. 10.17323/2587-814X.2026.3.69.89

[7] Мамедгулиев Р.И., Цуканова О.А., Брусакова И.А., Олейниченко А.В. Управление данными и интеграциями в процессе трансформации корпоративной архитектуры // Друкеровский вестник. 2026. № 69(1). С. 362–376. 10.17213/2312-6469-2026-1-362-376.

[8] Breiman L. Random Forest // Machine Learning. 2021. № 45(1). Pp. 5–32. 10.1023/A:1010933404324

[9] Gong Y., Janssen M. The value and myths about enterprise architecture // International Journal of Information Management. 2019. № 46. Pp. 1–9. 10.1016/j.ijinfomgt.20.18.11.006

[10] Kotusev S. TOGAF-based enterprise architecture practice: an exploratory case study // Communications of the Association for Information Systems. 2018. № 43(1). Pp. 321–359. 10.17705/1CAIS.04320

[11] Kotusev S., Kurnia S. The theoretical bases of enterprise architecture: A critical review and taxonomy of relevant theories // Journal of Information Technology. 2021. № 36(3). Pp. 275–315. 10.1177/0268396220977873

[12] Jonkers H., Lankhorst M., van Buuren R., Hoppenbrouwers S., Bonsangue M., van der Torre L. Concepts for modeling Enterprise Architecture // International Journal of Cooperative International Systems. 2004. № 13(3). С. 257–287. 10.1142/S0218843004000985

[13] Lankhorst M., van Buuren R., van Leeuwen D., Jonkers H., ter Doest H. Enterprise Architecture Modeling: The issue of integration // Advances Engineering Informatics. 2004. № 18(4). Pp. 205–216. 10.1016/j.aei.2005.01.005

[14] Martini A., Bosch J., Chaudron M.R.V. (2015). Investigating architectural technical debt accumulation and refactoring over time: A multiple-case study // Information and Software Technology. 2015. № 67. Pp. 237–253. 10.1016/j.infsof.2015.07.005

[15] Schmidt C., Buxmann P. Outcomes and success factors of enterprise IT architecture management: Empirical insight from the international financial services industry // European Journal of Information Systems. 2011. № 20(2). Pp. 165–185. 10.1057/ejis.2010.68

[16] Tamm T., Seddon P.B., Shanks G., Reynolds P. How does enterprise architecture add value to organizations? // Communications of the Association for Information Systems. 2011. № 28(1). Pp, 141–168. 10.17705/1CAIS.02810

[17] Tsoukalas D., Kehagias D., Siavvas M., Chatzigeorgiou A. Technical debt forecasting: an empirical studu on open-source repositories // Journal of Systems and Software. 2020. № 170. Art. 110777. 10.1016/j.jss.2020.110777