



Формирование замкнутого контура архитектурного управления корпоративными информационными системами

Васютин С.А., магистрант, Институт технологий управления,
МИРЭА — Российский технологический университет, Москва, Россия

Аннотация. В статье предложен подход к построению контура обратной связи между реализацией информационной системы и архитектурной моделью, основанный на использовании данных реверс-инжиниринга. Показано, что замкнутый контур объединяет инженерный, аналитический и управленческий уровни, обеспечивая переход от эпизодической актуализации архитектурной модели к регулярному и контролируемому процессу. Предложенный подход создает условия для повышения достоверности архитектурных данных и управляемости изменений, что отражается в качестве архитектурного контроля корпоративных информационных систем.

Ключевые слова: корпоративные информационные системы, архитектурное управление, реверс-инжиниринг, замкнутый контур управления

Forming a closed-loop architectural management framework for corporate information systems

Vasyutin S.A., Master's Student, Institute of Management Technologies,
MIREA — Russian Technological University, Moscow, Russia

Annotation. This article proposes an approach to constructing a feedback loop between information system implementation and the architectural model, based on the use of reverse engineering data. It is shown that the closed-loop framework integrates the engineering, analytical, and management levels, enabling the transition from episodic updating of the architectural model to a regular and controlled process. The

proposed approach creates conditions for increasing the reliability of architectural data and the manageability of changes, which is reflected in the quality of architectural control of corporate information systems.

Key words: corporate information systems, architectural management, reverse engineering, closed-loop management

Современные корпоративные информационные системы развиваются в условиях постоянного усложнения технологического ландшафта и высокой интенсивности изменений. Усложнение корпоративных информационных систем проявляется в увеличении количества приложений, одновременно используемых организациями. По данным ежегодных исследований *Businesses at Work*, среднее количество приложений на одну организацию увеличилось с 89 до 101. При этом в крупных организациях, численность сотрудников которых составляет не менее 2000 человек, среднее количество используемых приложений уже в исследовании 2022 г. достигало 187 единиц [5,6,9]. Представленная динамика свидетельствует о расширении корпоративного ИТ-ландшафта и увеличении числа объектов, состояние и взаимосвязи которых должны учитываться в процессе архитектурного управления (рис. 1).

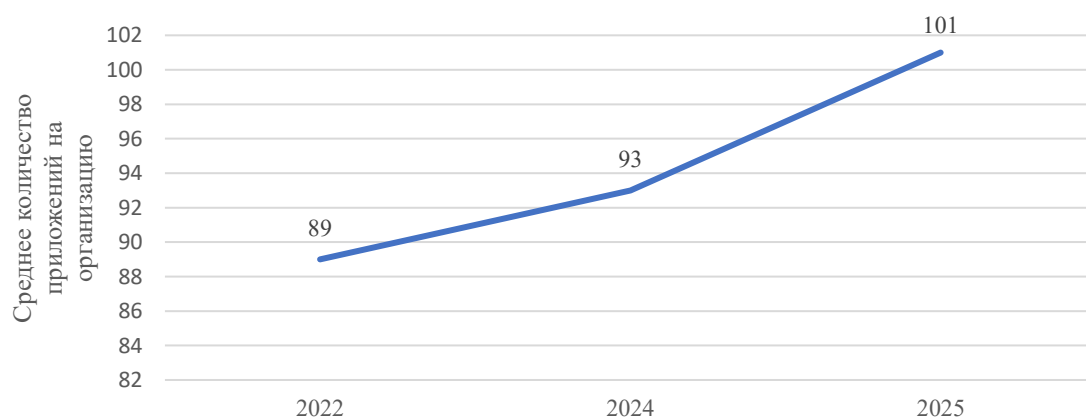


Рис. 1 – Динамика среднего количества приложений в корпоративном ИТ-ландшафте организации

Источник: составлено автором на основе [5,6,9]

Рост количества приложений в корпоративном ИТ-ландшафте организации не является одномерной характеристикой, приводящей к снижению качества архитектурного управления [3,11]. Однако такая тенденция существенно

повышает сложность поддержания актуального представления о корпоративной информационной системе (КИС). По мере расширения количества компонентов и зависимостей, возникает необходимость своевременно отражать текущие изменения в архитектурной модели [8,10]. Ручная и/или эпизодическая актуализация архитектурного описания создает риск накопления расхождений между фактическим состоянием информационных систем и их документированным представлением. Накопление разрыва снижает управляемость корпоративных информационных систем. Устаревшие сведения затрудняют анализ и повышают вероятность архитектурных отклонений, что приводит к снижению качества архитектурного контроля, параллельно увеличивая его трудоемкость. При наличии сотен информационных систем и тысяч выявленных расхождений поддержание архитектурной модели в актуальном состоянии перестает быть исключительно задачей документирования и становится самостоятельной проблемой управления сложным корпоративным ИТ-ландшафтом.

В этих условиях формирование механизма, обеспечивающего регулярную обратную связь между фактической реализацией информационной системы и ее архитектурным описанием, становится самостоятельным объектом управления. В отличие от традиционного нисходящего подхода, при котором движение осуществляется от бизнес-требований и архитектурных решений к технической реализации, современная модель архитектурного управления требует также встречного восходящего контура. Его назначение состоит в получении данных из фактических артефактов информационной системы и их сопоставление с документированной моделью для последующей экспертной оценке.

Цель исследования состоит в разработке подхода к формированию замкнутого контура архитектурного управления корпоративными информационными системами, обеспечивающего регулярное сопоставление фактического и документированного состояния ИС на основе данных реверс-инжиниринга.

Целевая модель замкнутого контура отражает переход от эпизодической актуализации архитектурной модели к регулярному и контролируемому процессу. Исходной точкой выступают релизные изменения, возникающие в результате развития автоматизированной системы. Изменения в жизненном цикле КИС возникают под воздействием большого количества триггеров от появления нового бизнес-требования до перенастройки безопасности или смены технологического стандарта.

Контур целевой модели процесса актуализации архитектурной модели КИС представлен на рис. 2.

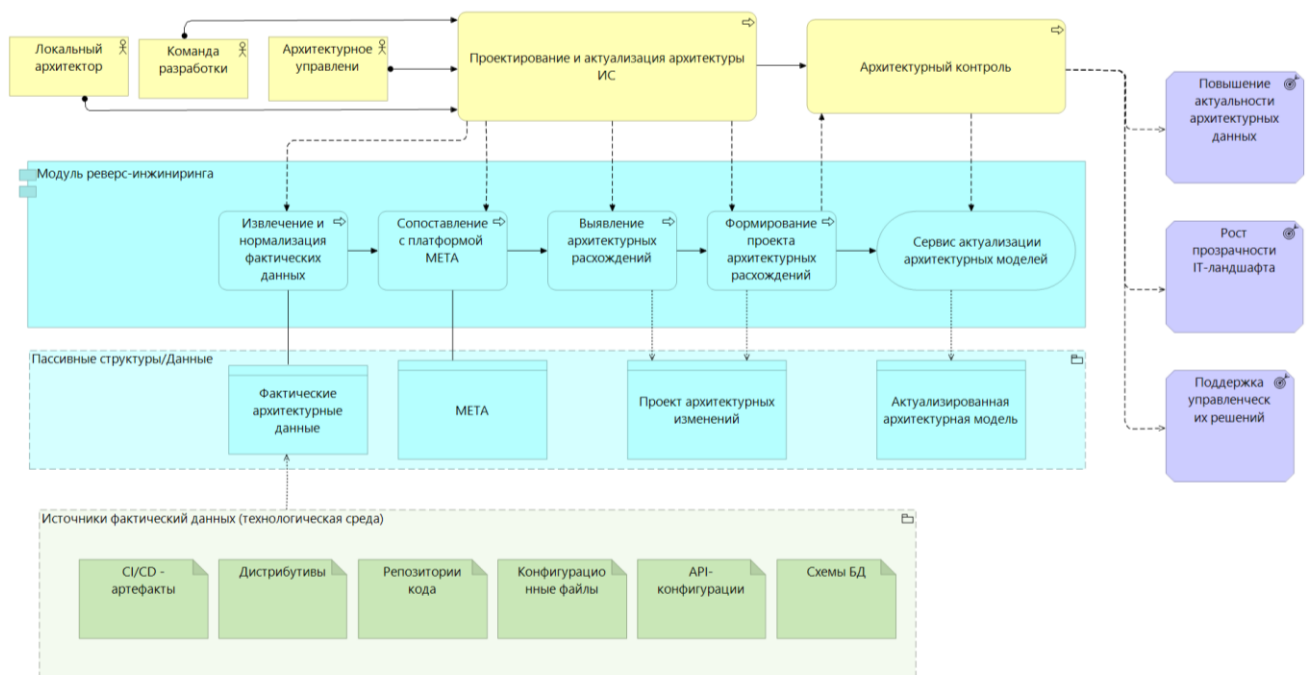


Рис. 2 – Целевая модель замкнутого контура на основе технологии реверс-инжиниринга (ArchiMate)

Источник: составлено автором

Первый блок целевой модели решает базовую задачу получения фактических данных из технических артефактов. Реверс-инжиниринг здесь нужен для того, чтобы увидеть реальный программный и конфигурационный состав системы, опираясь на ее описание [1,2]. Источниками анализа могут выступать: репозитории исходного кода, файлы зависимостей, спецификации API, схемы баз данных, конфигурационные файлы, контейнерные манифесты, материалы сборки, параметры развертывания и сведения о взаимодействии

сервисов. После обработки этих материалов появляется технический срез текущего состояния ИС, который выявляет реально задействованные компоненты, интерфейсы и интеграции, какие платформы используются и как выглядят данные и зависимости. Главная задача этого этапа состоит в том, чтобы выделить архитектурно значимую информацию. Технология реверс-инжиниринга извлекает те технические элементы системы, которые влияют на ее структуру, интеграционные связи, технологическую совместимость, эксплуатационную устойчивость, безопасность, масштабируемость и сопровождаемость. Именно поэтому в целевой модели нужны правила фильтрации и нормализации извлекаемых данных. Они позволяют уменьшать информационный шум и удерживать внимание архитектора на действительно значимых расхождениях.

Второй функциональный блок целевой модели связан уже с сопоставлением извлеченных фактических данных и документированного состояния ИС, представленного в корпоративной цифровой платформе, включающий реестр архитектурных данных (МЕТА). На этом этапе формируется дуополия «фактическое состояние - описанное состояние». Сравнение ведется по ключевым архитектурным объектам (сервисам, базам данных, интеграционным интерфейсам, информационным потокам, протоколам взаимодействия, средам развертывания, связанным архитектурным атрибутам, версиям и т.д.), по которым анализируется соответствие характеристик объектов и параметров реализации.

Третьим блоком выступает выявление и классификация расхождений. В целевой модели расхождения целесообразно группировать по нескольким типам:

1. отсутствующие в МЕТА элементы, но обнаруженные в фактическом состоянии ИС;
2. элементы, зафиксированные в МЕТА, но отсутствующие в фактическом состоянии. Скорее всего это кейсы, указывающие на устаревшие сведения архитектурной модели, которые требуют удаления, архивирования либо уточнения статуса.

3. несоответствия атрибутов, такие как различия в версиях, протоколах, параметрах взаимодействия, форматах данных, адресах сервисов, способах аутентификации и т.д.;

4. структурные расхождения, затрагивающие состав связей между компонентами и системами. Это сигналы о направлении и степени влияния изменений на формирование архитектурных рисков.

Классификация расхождений помогает сделать экспертную проверку более управляемой. Каждое расхождение при этом привязывается к источнику, из которого оно было получено. Благодаря такой трассируемости результат легче проверить, а риск субъективной трактовки становится ниже.

Следующий, четвертый блок целевой модели связан с формированием проекта архитектурных изменений, включающий набор предложений по актуализации архитектурной модели. Спектор таких изменений может быть весьма обширный от добавления нового объекта, до детализации и уточнения характеристики уже существующего элемента; от удаления устаревшей сущности до корректировки связи, фиксирования новой интеграции, обновления технологических атрибутов и т.д. Сам проект архитектурных изменений нужен как рабочий материал для локального архитектора. На этом шаге его роль становится не столько технической, сколько экспертной и управленческой. Вместо рутинной сверки он в большей степени занимается содержательной интерпретацией результатов реверс-инжиниринга.

Регистрация архитектурного отклонения означает, что выявленное состояние отличается от утвержденного архитектурного решения и потому требует отдельного контроля со стороны архитектурной функции. Архитектурный технический долг фиксируется в тех случаях, когда изменение принимается как временное решение, но при этом создает дополнительные ограничения для развития системы, усложняет сопровождение, усиливает зависимость от устаревших технологий или повышает риск дальнейшего масштабирования. За счет этого целевая модель связывает реверс-инжиниринг прежде всего с процессом управления качеством архитектурных решений.

Шестой блок целевой модели посвящен актуализации корпоративной цифровой платформы. После подтверждения архитектором проект архитектурных изменений переносится в архитектурный репозиторий.

В такой логике само появление релизного изменения становится поводом для запуска процедуры сопоставления, а архитектурный репозиторий начинает регулярно получать проверяемые данные из фактических источников. В результате процесс проектирования архитектуры становится более замкнутым. Проектное решение сначала фиксируется, затем реализуется, после этого проверяется по фактическим артефактам, сопоставляется с корпоративной цифровой платформой и при необходимости приводит к корректировке архитектурной модели.

За счет этого формируется более зрелая модель управления архитектурой ИС. Она объединяет проектное проектирование, фактическую инженерную реализацию и архитектурный контроль в один контур. В такой модели архитектура уже не выглядит статичным описанием, однажды зафиксированным на этапе согласования решения. Она регулярно уточняется на основании технических данных и потому становится динамической. Особенно это важно для микросервисных, интеграционно насыщенных и распределенных систем, где вручную отслеживать все изменения трудно из-за большого числа компонентов, высокой частоты релизов и параллельной работы нескольких команд.

При описанном подходе, меняется и архитектурный контроль. Если раньше он в основном был связан с проверкой проектных решений до реализации или с анализом уже подготовленных материалов, то при внедрении реверс-инжиниринга к нему добавляется пост-релизная верификация фактической реализации по отношению к архитектурной модели.

В целом работа с архитектурой корпоративной информационной системы построена по замкнутому циклу. Решение принимается и реализуется, после этого фактическое состояние системы проверяется через реверс-инжиниринг, проходит экспертную оценку и при необходимости отражается в корпоративной

цифровой платформе. Каждый новый проход делает картину системы яснее и со временем помогает заметить повторяющиеся проблемы.

Реверс-инжиниринг хорошо показывает компоненты, зависимости, интерфейсы, структуры данных, используемые технологии и конфигурации. Этого хватает, чтобы получить качественный технический срез. Но бизнес-смысл компонента, логика архитектурного решения, целевые ограничения, регуляторный контекст, критичность процесса и причины выбора технологии по-прежнему приходится читать и интерпретировать эксперту. Поэтому целевая модель не превращает архитектурное проектирование в обычный анализ кода. Она дает архитектору дополнительный фактический материал, который помогает поддерживать архитектурную модель в актуальном состоянии [7].

Замкнутый контур архитектурного управления можно рассматривать и в плоскости иерархического анализа уровней КИС. Первый уровень показывает состав системы, т.е. те конфигурационные элементы, которые реально существуют в действующей архитектурной модели (сервисы, модули, библиотеки, базы данных, очереди сообщений, внешние вызовы). Второй уровень выделяют для разбора интеграций. С его помощью легко отследить взаимодействие систем, а также актуальные протоколы, которые используются. Мониторингу подлежат направления операций: куда направлен обмен, какие форматы данных участвуют, где расположены точки входа.

Третий уровень относится к технологическому стеку. Здесь становятся заметны версии платформ, фреймворков, библиотек, контейнерных образов и инфраструктурных компонентов. Четвертый уровень связан с выявлением потенциальных архитектурных рисков. На этом уровне становятся заметны устаревшие зависимости, неописанные интерфейсы, несанкционированные связи, дублирование данных, нарушения принятых стандартов или появление технологических решений, выходящих за рамки целевой архитектуры.

Внедрение целевой модели предполагает настройку правил приоритизации выявленных расхождений (табл. 1).

Правила приоритизации изменений в архитектурной модели

Уровень приоритизации	Класс изменений
Высший приоритет	критичные бизнес-процессы; клиентские сервисы; межсистемные интеграции; персональные данные; финансовые операции; требования информационной безопасности; компоненты с высокой эксплуатационной нагрузкой
Средний приоритет	технологические атрибуты; версии компонентов и внутренние зависимости
Низкий приоритет	изменения, которые не влияют на архитектурную структуру или остаются в пределах допустимых параметров

Источник: составлено автором

Не все расхождения одинаково значимы, поэтому дифференцированный подход к работе с изменениями КИС помогает архитектору сосредоточиться на действительно существенных объектах контроля.

На следующем шаге формируется проект архитектурных изменений, который поступает архитектору на проверку. По итогам этой проверки одна часть изменений подтверждается, другая отклоняется, а часть переводится в статус уточнения или технического долга. Финальным состоянием становится обновленная архитектурная модель в корпоративной цифровой платформе и набор рассчитанных показателей эффективности [4].

В результате целевая модель объединяет три взаимосвязанных контура. Первый из них – инженерный, связанный с извлечением фактических данных из технических источников. Второй контур можно назвать аналитическим, он связан с сопоставлением данных, выявлением расхождений и формированием проекта архитектурных изменений. Третий контур уже относится к управленческому уровню. Здесь происходят экспертное подтверждение, актуализация корпоративной цифровой платформы, фиксация отклонений и расчет показателей. В совокупности эти контуры создают основу для повышения достоверности архитектурной модели и управляемости процесса проектирования архитектур КИС.

Предложенный подход предполагает, что целевая модель проектирования архитектур информационных систем с использованием реверс-инжиниринга развивается из действующей модели управления. Надстройкой будет являться

формирование регулярного контура обратной связи между фактическим состоянием ИС и репозиторием, который делает ИТ-ландшафт прозрачнее, что в свою очередь подготавливает почву для последующей оценки экономической эффективности внедрения.

Принципиальной особенностью предложенного контура является сочетание нисходящего и восходящего направлений архитектурного управления. Нисходящий контур сохраняет традиционную логику движения от требований и архитектурных решений к реализации, тогда как восходящий контур обеспечивает возвращение данных о фактическом состоянии системы в процесс архитектурного управления. За счет этого архитектурная модель получает возможность регулярно уточняться на основании проверяемых данных о фактически существующих компонентах и технологических характеристиках.

Библиографический список:

1. Белов Д. В. Развитие корпоративных информационных систем в контексте реинжиниринга бизнес-процессов // Вестник евразийской науки.– 2025. – Т. 17, № 1. –URL: <https://esj.today/PDF/52FAVN125.pdf> (дата обращения: 14.04.2026).
2. Израилов К. Е. Архитектура системы для проведения генетического реинжиниринга программы с поддержкой поиска разноуровневых уязвимостей // Вопросы кибербезопасности. – 2025. – № 1 (65). – С. 108–116.–DOI: 10.21681/2311-3456-2025-1-108-116.
3. Матраева Л. В. Промышленные бизнес-модели в эпоху Индустрии 4.0: от технологии к ценности / Л. В. Матраева, Е. С. Васютина // МИР (Модернизация. Инновации. Развитие). – 2025. – Т. 16, № 2. – С. 242-254. – DOI 10.18184/2079-4665.2025.16.2.242-254.
4. Ahn H., Kang S., Lee S. Reconstruction of an execution architecture view by identifying mapping rules for connectors // Journal of Systems and Software.– 2025.–Vol. 220.–Article 112268.–DOI: 10.1016/j.jss.2024.112268.
5. Bose A. Businesses at Work 2025: 10 years of data show how critical security has become / A. Bose // Okta. – 2025. – 12 March. – URL:

<https://www.okta.com/newsroom/articles/businesses-at-work-2025/> (дата обращения: 09.06.2026)

6. Businesses at Work 2022 Annual Report / Okta, Inc. – 2022. – 53 p. – URL: https://www.okta.com/content/dam/tmp---migration/files_live/2022-01/Businesses-at-Work-2022-Annual-Report.pdf (дата обращения: 09.06.2026).

7. Garcia J., Ivkovic I., Medvidovic N. A Comparative Analysis of Software Architecture Recovery Techniques // 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE). – Palo Alto : IEEE, 2013. – P. 486–496.

8. Kirschner Y. R., Gstür M., Sağlam T., Weber S., Koziolk A. Retriever: A View-Based Approach to Reverse Engineering Software Architecture Models // Journal of Systems and Software. – 2025. – Vol. 220. – Article 112277.

9. Nandy S. Businesses at Work 2024: Fundamentals in focus / S. Nandy // Okta. – 2024. – 7 March. – URL: <https://www.okta.com/newsroom/articles/businesses-at-work-2024/> (дата обращения: 09.06.2026).

10. Pattij M., van de Wetering R., Kurnia S., Kotusev S. Enhanced digital transformation supporting capabilities through enterprise architecture management // Digital Business. – 2022. – Vol. 2, № 1.

11. Rademacher F., Sachweh S., Zündorf A. A Modeling Method for Systematic Architecture Reconstruction of Microservice-Based Software Systems // Enterprise, Business-Process and Information Systems Modeling / ed. by S. Nurcan, I. Reinhartz-Berger, P. Soffer, J. Zdravkovic. – Cham : Springer, 2020. – Vol. 387. – P. 311–326.

References:

1. Belov D. V. Development of corporate information systems in the context of business process reengineering // Bulletin of Eurasian Science. – 2025. – Vol. 17, № s1. – URL: <https://esj.today/PDF/52FAVN125.pdf> (accessed: 14.04.2026).

2. Izrailov K. E. Architecture of a system for conducting genetic reengineering of a program with support for searching for multi-level vulnerabilities // Issues of cybersecurity. – 2025. – № 1 (65). – P. 108-116. – DOI: 10.21681/2311-3456-2025-1-108-116.

3. Matraeva L. V. Industrial business models in the era of Industry 4.0: from technology to value / L. V. Matraeva, E. S. Vasyutina // MIR (Modernization.

Innovation. Development). – 2025. – Vol. 16, № 2. – Pp. 242-254. - DOI 10.18184/2079-4665.2025.16.2.242-254.

4. Ahn H., Kang S., Lee S. Reconstruction of an execution architecture view by identifying mapping rules for connectors // Journal of Systems and Software. – 2025. – Vol. 220. – Article 112268. - DOI: 10.1016/j.jss.2024.112268.

5. Bose A. Businesses at Work 2025: 10 years of data show how critical security has become / A. Bose // Okta. – 2025. – 12 March. – URL: <https://www.okta.com/newsroom/articles/businesses-at-work-2025/> (access date: 06/09/2026)

6. Businesses at Work 2022 Annual Report / Okta, Inc. – 2022. – 53 p. – URL: https://www.okta.com/content/dam/tmp---migration/files_live/2022-01/Businesses-at-Work-2022-Annual-Report.pdf (access date: 06/09/2026).

7. Garcia J., Ivkovic I., Medvidovic N. A Comparative Analysis of Software Architecture Recovery Techniques // 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE). – Palo Alto: IEEE, 2013. – P. 486–496.

8. Kirschner Y. R., Gstür M., Sağlam T., Weber S., Kozirolek A. Retriever: A View-Based Approach to Reverse Engineering Software Architecture Models // Journal of Systems and Software. – 2025. – Vol. 220. – Article 112277.

9. Nandy S. Businesses at Work 2024: Fundamentals in focus / S. Nandy // Okta. – 2024. – 7 March. – URL: <https://www.okta.com/newsroom/articles/businesses-at-work-2024/> (access date: 06/09/2026).

10. Pattij M., van de Wetering R., Kurnia S., Kotusev S. Enhanced digital transformation supporting capabilities through enterprise architecture management // Digital Business. – 2022. – Vol. 2, № 1.

11. Rademacher F., Sachweh S., Zündorf A. A Modeling Method for Systematic Architecture Reconstruction of Microservice-Based Software Systems // Enterprise, Business-Process and Information Systems Modeling / ed. by S. Nurcan, I. Reinhartz-Berger, P. Soffer, J. Zdravkovic. – Cham: Springer, 2020. – Vol. 387. – P. 311–326.

Для цитирования: Васютин С.А., Формирование замкнутого контура архитектурного управления корпоративными информационными системами / Российский экономический интернет-журнал. – 2026. – № 3. URL:

© Васютин С.А., Российский экономический интернет-журнал 2026, № 3