



Ссылка для цитирования:

Шумилов К. А., Гурьева Ю. А. Моделирование сложных архитектурных объектов с использованием GRASSHOPPER, RHINO И ARCHICAD // Инженерно-строительный вестник Прикаспия : научно-технический журнал / Астраханский государственный архитектурно-строительный университет. Астрахань : ГАОУ АО ВО «АГАСУ», 2022. № 2 (40). С. 126–130.

УДК 519.685

DOI 10.52684/2312-3702-2022-39-1-130-136

УНАСЛЕДОВАННЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ. ПРОБЛЕМЫ И РЕШЕНИЯ

В. М. Зарипова, И. Ю. Петрова

Зарипова Виктория Мадияровна, кандидат технических наук, доцент кафедры систем автоматизированного проектирования и моделирования, Астраханский государственный архитектурно-строительный университет, г. Астрахань, Российская Федерация;

Петрова Ирина Юрьевна, доктор технических наук, заслуженный деятель науки Российской Федерации, профессор кафедры систем автоматизированного проектирования и моделирования, Астраханский государственный архитектурно-строительный университет, г. Астрахань, Российская Федерация; e-mail: irapet1949@gmail.com

В статье показан рост публикаций по вопросам сопровождения унаследованных информационных систем, что подтверждает актуальность темы исследования. Дано базовое представление об опасности эксплуатации таких информационных систем и программного кода. Показано, чем эти системы могут навредить бизнесу. Представлена систематизация возникающих проблем и возможных решений по сопровождению унаследованных информационных систем. Сделана попытка систематизировать терминологию в этой области. На основе анализа большого количества отечественной и зарубежной литературы авторами показаны тренды в области сопровождения и модификации унаследованных информационных систем – это использование облачных технологий и переход к микросервисной архитектуре.

Ключевые слова: информационная система (ИС), унаследованные ИС, миграция, реинжиниринг, модернизация, микросервисы, облачные технологии.

LEGACY INFORMATION SYSTEMS. PROBLEMS AND SOLUTIONS

V. M. Zaripova, I. Yu. Petrova

Zaripova Viktoriya Madiyarovna, Candidate of Technical Sciences, Associate Professor of the Department of Computer-Aided Design and Modeling Systems, Astrakhan State University of Architecture and Civil Engineering, Astrakhan, Russian Federation;

Petrova Irina Yurievna, Doctor of Technical Sciences, Honored Worker of Science of the Russian Federation, Professor of the Department of Computer-Aided Design and Modeling Systems, Astrakhan State University of Architecture and Civil Engineering, Astrakhan, Russian Federation; e-mail: irapet1949@gmail.com

The article shows the growth of publications on maintenance of legacy information systems, which confirms the relevance of the research topic. Authors give basic understanding of the risk of exploitation of legacy information systems and program code and show how such systems may bring harm to business. A systematization of emerging problems and possible solutions for maintaining legacy information systems is presented. An attempt was made to systematize the terminology in this area. Based on the analysis of a large number of domestic and foreign literature, the authors show trends in the field of maintaining legacy information systems - this is the use of cloud technologies and the transition to microservice architecture.

Keywords: information system (IS), legacy IS, migration, reengineering, modernization, microservices, cloud technologies.

Введение

Несмотря на дорогостоящие и инновационные механизмы цифровой трансформации, многие организации по-прежнему используют и поддерживают устаревшее программное обеспечение (ПО). Это связано с тем, что с годами это ПО стало основой функционирования организации, и заменить его современными альтернативами непросто. Длительный период использования делает такие системы практически незаменимыми для организации, поскольку они поддерживают важную бизнес-логику и обеспечивают данные, необходимые в повседневных бизнес-операциях.

В отчете VMware и MIT Technology Review за декабрь 2017 г. говорится, что 62 % ИТ-руководителей назвали интеграцию устаревших систем своей самой большой проблемой при переходе к облачным технологиям [1].

Однако, модернизация унаследованных систем и ИТ-решений – это то, что многие предприятия просто обязаны сделать в рамках достижения более амбициозных бизнес-целей. Только модернизация может повысить совместимость и интегрируемость унаследованных систем в новую инфраструктуру, обеспечить сетевую безопасность, добавить новые функции и снизить стоимость обслуживания.

Информационная система, которая основана на устаревших технологиях, но имеет решающее значение для повседневных операций называется «унаследованная или устаревшая система» (*legacy system*). Этот термин начали использовать, начиная с 70-х гг. прошлого века.

Унаследованная система – это «информационная система, которая может быть основана на устаревших технологиях, но имеет решающее значение для повседневных операций» [2].

Срок действия большинства программ составляет от 10 до 15 лет [3]. Однако по данным Счетной палаты правительства США возраст десяти важнейших устаревших систем федеральных информационных технологий, которые больше всего нуждаются в модернизации, составляет от 8 лет до 51 года, и в совокупности их эксплуатация и обслуживание обходятся примерно в 337 млн долл. в год [4].

Несколько десятилетий назад предприятия активно использовали ряд распространенных технологий, которые сейчас признаны устаревшими (мейнфреймы, языки программирования COBOL или Fortran, операционные системы MS-DOS, Windows 3.1 или XP и т. д.). Эти системы и до сих пор используются некоторыми предприятиями и компаниями из-за предшествующих крупных инвестиций, страха перед изменениями, непрерывности операций и трудностей с внедрением новых технологий. Кроме того, эти системы по-прежнему отвечают потребностям бизнеса, и сотрудники хорошо ими владеют [5, 6]. С момента внедрения подобных систем в их хранилищах накоплено много важной бизнес-информации, в формате не поддерживающим или затрудняющим конвертацию [7]. Такие системы препятствуют совершенствованию, которое необходимо организациям для соответствия глобальным изменениям. Процесс обслуживания затруднен из-за высокой стоимости и отсутствия специалистов, разбирающихся в этих системах, а также из-за отсутствия документации [8, 9]. Эти системы также трудно расширять и интегрировать с новыми информационными системами из-за ограничений старой технологии [10]. В результате инновационные усилия организации часто сдерживаются из-за этих систем [11].

В настоящее время из-за широкого внедрения современных цифровых технологий в различные сферы жизни и производства модернизация устаревших систем имеет первостепенное значение для предоставления пользователям конкурентоспособных и инновационных продуктов и услуг [12]. Модернизация позволяет использовать новые/прорывные технологии, такие как искусственный интеллект, высокопроизводительные вычисления, облачные вычисления, Интернет вещей, робототехника и большие данные [13].

Однако решение о необходимости модернизации практически повсеместно принимается или отклоняется не ИТ отделом, а руководящими менеджерами или ведущими специалистами. Как правило, в расчет при этом принимается только стоимость такой модернизации.

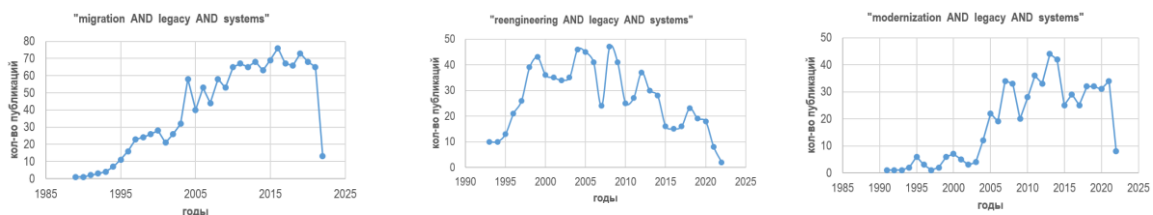
Цель данной статьи – дать базовое понимание специалистам, далеким от разработки, о том, чем опасны устаревшие информационные системы и программный код, чем такие системы могут навредить бизнесу, а также представить систематизацию возможных решений по сопровождению устаревших информационных систем.

Обзор литературы

Анализ публикаций по данной теме в БД Scopus был проведен по трем поисковым запросам, результаты запросов показаны на рисунке 1. Изменение количества публикаций по годам показано на рисунке 2.

3	TITLE-ABS-KEY (migration AND legacy AND systems)	Результатов: 1,356
2	TITLE-ABS-KEY (reengineering AND legacy AND systems)	Результатов: 810
1	TITLE-ABS-KEY (modernizing AND legacy AND systems)	Результатов: 127

Рис. 1. Результаты поисковых запросов в БД Scopus



А) Результат запроса «migration AND legacy AND systems» Б) Результат запроса "reengineering AND legacy AND systems" В) Результат запроса "modernization AND legacy AND systems"

Рис. 2. Рост количества публикаций по проблемам модернизации и реинжиниринга устаревшего программного обеспечения

Как следует из анализа этих графиков, количество публикаций, посвященных проблемам реинжиниринга, модернизации и миграции унаследованных ИС возрастает. Поэтому важно, чтобы исследования по сопровождению таких программных систем и их модернизации были тщательно изучены.

Анализ этих публикаций позволил выявить и классифицировать проблемы и возможные решения по сопровождению устаревших программных систем и их модернизации, что будет рассмотрено в следующих разделах этой статьи.

В России проблеме модернизации унаследованного ПО также посвящено большое количество

работ. Так, в [14] дано определение понятия реинжиниринга как процесса перепроектирования бизнес-процессов и связанных с ними систем и организационных структур для достижения значительного улучшения эффективности бизнеса.

В работах [15, 16] предложено использовать системный анализ при реинжиниринге унаследованного программного обеспечения.

В работах [17, 18] показана особая важность перехода от устаревшего ПО импортного производства к отечественным разработкам для военно-промышленного комплекса России в усло-

виях нарастающих политических и экономических санкций со стороны США, Европейского союза, Украины и других стран.

Вопросы терминологии

В российской и зарубежной литературе используется несколько близких по смыслу понятий: реинжиниринг ИС [14–16, 19], модернизация ИС [6, 7, 11–13], миграция ИС [17, 24].

Анализируя эти источники, можно сделать вывод, что терминология в настоящее время еще не устоялась. Авторы в России и за рубежом используют близкие по смыслу термины для описания одних и тех же процессов, методов, технологических решений.

В данной работе предлагаются следующие определения.

Реинжиниринг ИС (reengineering) – это фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов с целью использования информационных технологий для повышения производительности и сокращения затрат. В работах [14, 15] представлен подробный обзор терминологии в области реинжиниринга ИС.

Реинжиниринг бизнес-процессов и соответственно устаревшей ИС – наиболее радикальный, практичный и эффективный подход, однако он трудоемок и требует осознанной поддержки руководства. Реинжиниринг ИС позволяет исследовать и модифицировать устаревший программный код, что помогает поддерживать, повторно использовать и изменять его в будущем. Затраты могут быть очень высокими, поэтому важно правильно оценить и спланировать проект преобразования такой системы. Объектами реинжиниринга могут являться функциональная, информационная и/или архитектурная модели информационной системы.

Модернизация ИС (modernization) – это масштабные изменения, которые заменяют всю систему и необходимы, если приложение больше не соответствует бизнес-требованиям. В рамках модернизации системы все компоненты перерабатываются и развиваются. Можно сказать, что модернизация ИС – это технологически эффективный и экономически оправданный способ улучшения устаревших систем.

Миграция ИС (migration) – перемещает систему в новую техническую среду, потенциально включающую управление данными, связь и среду программирования [16, 20]. Миграция является частным случаем модернизации. В процессе миграции могут быть пересмотрены функциональные модули системы, ее архитектура, осуществлен рефакторинг кода. Так как миграция может затрагивать ряд различных компонентов ИС, то этот термин может использоваться как для преобразования ИС в целом, так и для отдельных ее компонентов (программное обеспечение, данные, оборудование, приложения).

Проблемы унаследованных систем

За время своей эксплуатации любая система накапливает большой объем данных и знаний, необходимый для эффективного и точного выполнения критически важных бизнес-задач. Поэтому обслуживание унаследованных программных систем и миграция таких данных и знаний имеет приоритетное значение во многих организациях. И несмотря на эти сопутствующие трудозатраты организации продолжают штатно эксплуатировать подобные системы из принципа «лучшее – враг хорошего».

Тем не менее устаревшие программные системы имеют ряд недостатков, которые рассмотрены ниже.

- **высокая стоимость обслуживания** – эксплуатация унаследованной ИС осуществляется, как правило, на старых компьютерах, поэтому требуются значительные средства для поддержки того оборудования в работоспособном состоянии. Например, правительство США ежегодно тратит более 100 млрд долл. на инвестиции в ИТ и киберпространство. Из этой суммы около 80 % затрачивается на эксплуатацию и обслуживание существующих инвестиций в ИТ, включая устаревшие системы [4];

- **высокая стоимость обновлений** – новая лицензионная политика мировых брендов ПО направлена на продажу лицензии только на один год или перевод платных сервисов в облако. Как показывает практика, даже крупные компании вынуждены переходить на ПО с открытым кодом. В качестве примера можно привести известный факт, что компания «Яндекс» в течении нескольких лет планомерно переходила на популярную свободную СУБД PostgreSQL [21];

- **неподдерживаемое ПО** – не поддерживается разработчиком и из-за обновления оперативной системы перестает работать. Если ПО действительно актуально и необходимо, то его стоит модернизировать для совместимости с новой версией ОС;

- **эффект производительности** – долгое время загрузки, задержка в работе компьютерного приложения и т.д. связано с тем, что производители больше не выпускают обновления, нет реального способа решить проблемы;

- **устаревание документации на поддерживаемые бизнес-процессы** – при изменении требований и доработке ПО длительное время никто не вносил изменений в документацию по программе. В результате документация не соответствует действительности и любые изменения ПО несут риски негативного воздействия на бизнес-процессы;

- **старые сотрудники уходят вместе со своими знаниями** – в организации, эксплуатирующей устаревшее программное обеспечение может полностью или частично отсутствовать документация, руководства для пользователей и администраторов системы, а старые сотрудники и разработчики этой ИС были уволены,

ушли на пенсию и т. д. У новых сотрудников возникают трудности в процессе эксплуатации, унаследованной ИС, так как они не могут разобрататься в принципах ее функционирования;

- **повышенный риск безопасности** – старые операционные системы без исправлений безопасности подвержены взлому, вирусам и другим вредоносным воздействиям. Надежность встроенной защиты уменьшается с течением времени, потому что злоумышленники изучили принципы защиты системы и могут их обходить. Иногда единственный способ защитить такую систему – это изолировать ее от остальной части предприятия, что может оказаться нецелесообразным, если необходимо взаимодействие с другими системами. Например, атака программы-вымогателя WannaCry в мае 2017 г. выявила уязвимость устаревших систем, которые больше не поддерживаются Microsoft. Более 500 000 компьютеров, работающих под управлением Windows XP, Windows 8 и Server 2003, пострадали в 200 странах из-за отсутствия обновлений безопасности [22];

- **проблема интеграции с внешними системами** – из-за различий в используемых базовых технологиях, невозможно интегрироваться с ведущими сервисами, своевременно масштабировать или гибко модифицировать ПО. В результате развитие бизнеса замедляется, организация становится неконкурентоспособной. Особенно четко этот симптом наблюдается в государственных учреждениях, отличающихся громоздкой и негибкой структурой, где любые изменения требуют долгосрочного согласования. Например, чрезмерные затраты правительства США на ИТ инфраструктуру в 2019 г. (более 70 миллиардов долларов) послужили причиной внутреннего расследования, которое показало, что основной проблемой является именно устаревшее ПО [23];

- **удобство использования** – пользователей не волнует, основана ли система на старом методе, технологии или использует старое оборудование. Пользователям нужны надежные и безопасные системы, более простой пользовательский интерфейс, который не требует изучения каких-либо команд для работы с программным обеспечением;

- **дублирование и потери данных** – эта проблема связана с трудностями интеграции устаревших систем, которые функционируют в организации изолированно друг от друга. Это приводит к изолированности информации. Сотрудники должны использовать множество различных инструментов, чтобы найти нужные данные;

- **невозможно использовать технологии больших данных** – исторические данные не преобразованы в новый формат и могут использоваться в новой системе только с применением специальных таблиц перевода или требуют использования специализированных хранилищ данных;

- **политические причины** – также могут воздействовать на принятие решения о миграции

системы. Например, в результате введения антироссийских санкций многие зарубежные поставщики программного обеспечения ушли с российского рынка, что делает жизненно необходимым переход на российское ПО.

На рисунке 3 показаны проблемы и возможные решения по сопровождению устаревших программных систем и их модернизации.

УСТАРЕВШИЕ СИСТЕМЫ	
ПРОБЛЕМЫ	ВОЗМОЖНЫЕ РЕШЕНИЯ
<ul style="list-style-type: none"> ➤ Высокая стоимость обслуживания ➤ Высокая стоимость обновлений ➤ Нетоддерживаемое ПО ➤ Эффект производительности ➤ Устаревание документации на поддерживаемые бизнес-процессы ➤ Старые сотрудники уходят вместе со своими знаниями ➤ Повышенный риск безопасности ➤ Проблема интеграции с внешними системами ➤ Удобство использования ➤ Дублирование и потери данных ➤ Невозможно использовать технологии больших данных ➤ Политическая причина 	<ul style="list-style-type: none"> ➤ Инкапсуляция (Encapsulate) - доступ в слои через API ➤ Перенос на другую платформу - развёртывание устаревшего приложения в другой физической, виртуальной или облачной инфраструктуре <ul style="list-style-type: none"> • Рестейлинг (Rebelling) - перенос локальных приложений в облако без внесения серьезных изменений ➤ Резаплаформинг (replatforming) - части кода могут быть переставлены или может быть изменен способ взаимодействия приложения с базой данных. ➤ Рефакторинг (refactoring) - реструктуризация программного кода без изменения его первоначальной функциональности ➤ Редизайн (redesign) - творческая работа пользователейских интерфейсов ➤ Перепроектирование (Rebuilding) - модернизация ИС с изменением архитектуры и программного кода. ➤ Замена (Replacing) - полная замена ИС

Рис. 3. Проблемы и возможные решения по сопровождению устаревших программных систем и их модернизации

Анализ возможных решений по сопровождению устаревших информационных систем

Жизненный цикл ИС состоит из шести этапов: технико-экономическое обоснование, анализ системы, проектирование системы, программирование и тестирование, установка, эксплуатация и техническое обслуживание. Если ИС перестает удовлетворять требованиям после нескольких лет эксплуатации, то она заменяется на новую систему (рис. 4).

Модернизация ИС - эволюционный этап глобального жизненного цикла ИС без изменений основных функциональных возможностей. Во время модернизации системы некоторые модули могут быть сохранены. Эти модули называются унаследованными модулями, они не изменяются, не адаптируются, доступ к ним происходит через интерфейсы. Этап модернизации тесно связан с сопровождением системы [13].

На рисунке 4 приведен жизненный цикл информационной системы с модернизацией. Первоначальная система (ИС1) была спроектирована в соответствии с поставленными бизнес-требованиями. Дальнейшее изменение бизнес-требований вызывает эволюцию ИС1. *Эволюция (evolution) ИС* – это небольшие изменения, которые вносятся время от времени и адаптируют функции, интерфейсы, хранилище данных и так далее к изменившимся требованиям. Критические изменения требований, либо эффект накопленных изменений требуют модернизации ИС1, после чего эволюция ИС продолжается до момента, когда изменение унаследованной ИС1, по каким-то причинам, уже не позволяет решить поставленных задач. Происходит замена ИС1 на ИС2 с миграцией необходимых данных и процесс повторяется для ИС2.

Необходимо понимать, что вертикальная ось функциональности ИС отражает не наращивание функциональности ИС как таковой, а степень соответствия этой функциональности поставленным бизнес-целям. В случае изменения бизнес-целей, может потребоваться полный реинжиниринг всех процессов и систем.

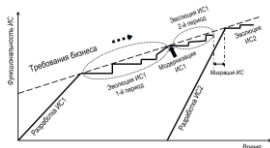


Рис. 4. Жизненный цикл информационной системы

Виды возможных решений по сопровождению унаследованного программного обеспечения показаны на рисунке 5.

Инкапсуляция (encapsulate) – это один из самых экономичных и быстро реализуемых методов повторного использования устаревших программных компонентов. Оставляя программный код и данные в текущей среде, инкапсуляция предоставляет доступ к слоям через API, т. е. просто обеспечивает новый интерфейс для текущей устаревшей системы. Это помогает использовать приложение и расширять его возможности и ценность, но не решает проблемы обслуживания. Этот вариант целесообразно использовать, если устаревшее приложение имеет качественный код, высокую ценность для бизнеса и поддерживает требуемую функциональность [24], но интерфейсы приложения перестали отвечать требованиям бизнеса (непривлекательны, устарели, не поддерживают современные способы обмена информацией и т. д.) или неудобны для использования.



Рис. 5. Виды решений по сопровождению унаследованного программного обеспечения

Перенос на другую платформу (rehosting и re-platforming) – повторное развертывание устаревшего приложения в другой физической, виртуальной или облачной инфраструктуре без изменения или с минимальными изменениями кода или функциональности.

Рехостинг (rehosting) – самый популярный способ модернизации из-за его простоты и доступности. Под рехостингом понимается простое копирование и вставка локальных приложений в облако без внесения серьезных изменений. Миграция приложения без каких-либо модификаций позволяет ускорить процесс модернизации с минимальными затратами ресурсов. При переносе в облако используются базовые облачные функции.

Реплатформинг (replatforming) – некоторые части кода могут быть переписаны или может быть изменен способ взаимодействия приложения с базой данных. Все изменения вносятся с целью наибольшего соответствия приложения новой платформе (как правило облачной среде) и максимального использования ее функций.

Миграция в облако (Cloud migration) – миграция устаревших программных систем в облако, в том числе включая переход на сервис-ориентированную архитектуру (Service – Oriented Architectur = SOA) или

микросервисную архитектуру является одним из инновационных путей модернизации унаследованных систем [1, 25]. Она включает перемещение данных, приложений и других цифровых активов бизнеса из локальной инфраструктуры на сервер виртуального компьютера с общими ресурсами (рис. 6). Миграция устаревших систем в облако имеет ряд преимуществ: экономия затрат, масштабируемость, мобильность и безопасность.

В 2020 и 2021 гг. произошел взрывной рост облачных вычислений, так как многие предприятия вынуждены были перевести сотрудников на удаленную работу на фоне глобальной пандемии. В свою очередь это привело к росту различных цифровых сервисов. По данным Gartner, к 2025 г. более 95 % новых цифровых рабочих нагрузок будут развернуты на облачных платформах по сравнению с 30 % в 2021 г. [26]. Облачные вычисления могут снизить затраты на оплату труда ИТ-специалистов на 50 % [27]. Устаревшие системы можно перенести с помощью множества инструментов интеграции ETL (Extract, Transform, Load), которые служат для извлечения, преобразования и загрузки данных в новое хранилище или в другую базу. Эти инструменты помогают пользователям извлекать данные из устаревших систем и передавать их в промежуточную область. В промежуточной области данные очищаются и подготавливаются к окончательной миграции в облако.



Рис. 6. Миграция устаревших программных систем на сервис-ориентированную архитектуру

Рефакторинг (refactoring) – процесс реструктуризации программного кода без изменения его первоначальной функциональности, аппаратного обеспечения или структуры приложения в целом. [28]. Цель рефакторинга — сделать код программы более легким для понимания и улучшить его путем внесения множества небольших изменений без изменения внешнего поведения кода. Методы рефакторинга приведены в [29]. Отличие рефакторинга от реинжиниринга состоит в том, что последний осуществляют для расширения функциональности программного обеспечения. Рефакторинг позволяет более эффективно поддерживать существующий проект и снизить затраты на его модернизацию. Как правило, крупные рефакторинги предваряют реинжиниринг ИС.

Другая разновидность рефакторинга – изменение архитектуры программной системы [30]. Для предприятий, уже работающих в облаке, это идеальный вариант перехода с контейнеров на микросервисы, чтобы повысить масштабируемость и изоляцию компонентов.

Согласно прогнозу IDC к 2022 г. 90 % всех новых приложений будут иметь *микросервисную архитектуру*, которая улучшит возможности проектирования, отладки, обновления и использования стороннего кода [31].

Многие современные устаревшие приложения реализованы как монолиты, в которых все компоненты, модули имеют сильную связность между собой, а все функции для всех объектов данных обрабатываются на сервере.

При переходе на микросервисную архитектуру расширяется масштабируемость системы. Наиболее очевидным это преимущество становится при преобразовании в микросервисы тех компонентов, которые используются большинством пользователей. В работах [32, 33] подробно описан план миграции унаследованных приложений на микросервисы. Возможно построение унифицированной модели данных на основе онтологии, которая позволяет не только описывать сложные связи между сущностями, но и упрощает процесс взаимодействия и интерпретации данных от одного микросервиса к другому, т. е. упрощает интеграцию между сервисами [34].

Редизайн (redesign) – переработка только пользовательских интерфейсов информационной системы без существенного вмешательства в ее устройство, обычно применим к веб-сайтам, где большое значение имеет эргономика веб-интерфейса или юзабилити [35, 36]. Это процесс, направленный на улучшение взаимодействия с пользователем, в первую очередь путем переоценки существующей навигации, макетов, контента и стека технологий [37].

Перепроектирование (Rebuilding) – переписывают компоненты приложения с нуля, сохраняя при этом их объем и спецификации. Это более продвинутый подход к модернизации устаревшей системы, позволяющий создать новую архитектуру и внести изменения в код. Перепроектирование приложения позволяет включать новые функции и процессы с использованием возможностей современных технологий и платформ. Этот подход также используется

для решения проблем с производительностью и масштабируемостью ИС.

Замена (Replacing) – полная замена приложения другой ИС. В этом случае повторное использование существующей устаревшей бизнес-логики невозможно, так как она больше не отвечает потребностям бизнеса или стратегическим целям предприятия. Метод замены используется, когда существующая система не может удовлетворить все требования организации бизнес-процесса.

Преимущества полной замены устаревшей ИС:

- введение большего количества цифровых возможностей, которые ранее были недоступны;
- повышение надежности и производительности;
- большая маневренность и гибкость в удовлетворении меняющихся бизнес-требований и запросов пользователей;
- повышение привлекательности для молодых ИТ сотрудников, которые хотят осваивать новые технологии;
- снижение капитальных затрат, связанных с использованием облачных технологий вместо локальных.

Выводы

- Количество публикаций по проблемам модернизации унаследованных информационных систем возрастает с каждым годом, т. е. эта проблема является актуальной.
- До сих пор нет сложившейся терминологии в области сопровождения устаревших информационных систем. В данной статье сделана попытка классификации видов решений по сопровождению унаследованного программного обеспечения.
- Показано, что тренды в области сопровождения устаревших информационных систем – это использование облачных технологий и изменение их архитектуры.
- Проанализированы подходы к эволюции, миграции и модернизации устаревших программных систем и основания для выбора того или иного подхода.

Список литературы

1. 62 % of IT leaders say legacy systems are biggest roadblock to multi-cloud success // URL: <https://www.techrepublic.com/article/62-of-it-leaders-say-legacy-systems-are-biggest-roadblock-to-multi-cloud-success/> (дата обращения 15.03.2022).
2. Gartner Glossary. Legacy Application Or System, URL: <https://www.gartner.com/en/information-technology/glossary/legacy-application-or-system> (дата обращения 21.03.2022).
3. Mallidi, Ravi Kiran & Sharma, Manmohan & Singh, Jagjit. (2021). Legacy Digital Transformation: TCO and ROI Analysis. International journal of electrical and computer engineering systems. 12. 163-170. 10.32985/ijeces.12.3.5.
4. GAO, Information Technology: Agencies Need to Develop and Implement Modernization Plans for Critical Legacy Systems, 2021, <https://www.gao.gov/products/gao-21-524t> (дата обращения 24.03.2022).
5. Jha, S.; Jha, M.; O'Brien, L.; Wells, M. Supporting Decision Making with Big Data Integrating Legacy Systems and Data. In Proceedings of the 2017 4th Asia-Pacific World Congress on Computer Science and Engineering, APWC, Mana Island, Fiji, 11–13 December 2017; pp. 120–128.
6. Abu Bakar, H.; Razali, R.; Jambari, D.I. Legacy Systems Modernisation for Citizen-Centric Digital Government: A Conceptual Model. Sustainability 2021, 13, 13112. <https://doi.org/10.3390/su132313112>.
7. Seetharamantray, H.; Murulidhar, N.; Chandrasekaran, K. Implications of Legacy Software System Modernization—A Survey In A Changed Scenario. Int. J. Adv. Res. Comput. Sci. 2017, 8, 1002–1008.
8. D. Beach, Legacy systems push up costs of digital transformation, 2019, <https://www.theglobaltreasurer.com/2018/09/27/legacy-systems-push-up-costs-of-digital-transformation/>.
9. Crotty, J.; Horrocks, I. Managing Legacy System Costs: A Case Study of a Meta-Assessment Model to Identify Solutions in a Large Financial Services Company. Appl. Comput. Inform. 2017, 13, 175–183.
10. Srinivas, M.; Ramakrishna, G.; Rajasekhara Rao, K.; Suresh Babu, E. Analysis of Legacy System in Software Application Development: A Comparative Survey. Int. J. Electr. Comput. Eng. 2016, 6, 292–297.

11. Assunção, Wesley. (2021). Contemporary Software Modernization: Perspectives and Challenges to Deal with Legacy Systems. 10.13140/RG.2.2.25176.42243.
12. D. Wolfart, W. K. G. Assunc, ao, I. F. da Silva, D. C. P. Domingos, E. Schmeing, G. L. D. Villaca, and D. d. N. Paza, "Modernizing legacy systems with microservices: A roadmap," in 25th Evaluation and Assessment in Software Engineering (EASE). ACM, 2021, p. 149–159.
13. P. L. Leon and F. E. A. Horita, "On the modernization of systems for supporting digital transformation: A research agenda," in XVII Brazilian Symposium on Information Systems, 2021, pp. 1–8.
14. Ахтырченко К.В., Сорокваша Т.П. Методы и технологии реинжиниринга ИС // Труды ИСП РАН. 2003. №. URL: <https://cyberleninka.ru/article/n/metody-i-tehnologii-reinzhiniringa-is> (дата обращения: 16.03.2022).
15. Массель Л.В., Подкаменный Д.В. Системный анализ и реинжиниринг унаследованного программного обеспечения // Наука и образование (электронное научно-техническое издание МГТУ им. Баумана, эл. № ФС 77 – 30569. Гос. рег. № 0421000025). №4. 2011. Режим доступа: <http://technomag.edu.ru> (дата обращения: 02.12.2021).
16. Массель А.Г. Адаптация методики реинжиниринга унаследованных программных систем / А. Г. Массель, Т. Г. Мамедов // Информационные и математические технологии в науке и управлении. – 2021. – № 4(24). – С. 88-99. – DOI 10.38028/ESI.2021.24.4.009.
17. Трошков С.Н. Об опыте миграции приложений на свободно распространяемое программное обеспечение с открытым кодом // Вестник НГУ. Серия: Информационные технологии. 2018. №2. URL: <https://cyberleninka.ru/article/n/ob-opyte-migratsii-prilozheniy-na-svobodno-rasprostranyaemoe-programmnoe-obespechenie-s-otkrytym-kodom> (дата обращения: 16.03.2022).
18. Братченко А.И., Бутусов И.В., Романов А.А. О проблемах обеспечения технологической независимости предприятий оборонно-промышленного комплекса // Военная мысль. 2018. №6. URL: <https://cyberleninka.ru/article/n/o-problemah-obespecheniya-tehnologicheskoy-nezavisimosti-predpriyatij-oboronno-promyshlennogo-kompleksa> (дата обращения: 16.03.2022).
19. Held Back by Legacy Software? Reengineering Can Help To Fuel Change, URL: <https://www.scnsoft.com/blog/legacy-software-reengineering-guide>, (дата обращения 25.03.2022).
20. A. Sivagnana Ganesan, T. Chithralekha, 2017, A Comparative Review of Migration of Legacy Systems, International journal of engineering research & technology (IJERT) Volume 06, Issue 02 (February 2017), <http://dx.doi.org/10.17577/IJERTV6IS020319>.
21. «Яндекс»: Переход с Oracle на открытое ПО, URL: [https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82:%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81_\(%D0%BC%D0%B8%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F.%D1%81_Oracle_%D0%BD%D0%B0_PostgreSQL\)](https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82:%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81_(%D0%BC%D0%B8%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F.%D1%81_Oracle_%D0%BD%D0%B0_PostgreSQL)) (дата обращения 17.03.2022).
22. Интерфакс, Создатели вируса WannaCry получили в качестве выкупа \$42 тысячи, URL: <https://www.interfax.ru/world/562284>, (дата обращения 24.03.2022).
23. U.S. Government Accountability Office (GAO), Information Technology: Agencies Need to Develop Modernization Plans for Critical Legacy Systems, URL: <https://www.gao.gov/products/gao-19-471#summary> (дата обращения 17.03.2022).
24. Legacy System Modernization: When and How to Do It Right, URL: <https://euristiq.com/legacy-system-modernization-when-and-how-to-do-it-right/>, (дата обращения 24.02.2022).
25. Abdellatif, Manel & Shatnawi, Anas & Mili, Hafedh & Moha, Naouel & El-Boussaidi, Ghizlane & Hecht, Geoffrey & Privat, Jean & Guéhéneuc, Yann-Gaël. (2020). A taxonomy of service identification approaches for legacy software systems modernization. Journal of Systems and Software. 10.1016/j.jss.2020.110868.
26. Top cloud migration and adoption trends for 2022 and beyond, URL: <https://www.techtarget.com/searchcloudcomputing/feature/Follow-these-top-cloud-migration-trends>, (дата обращения: 28.03.2022).
27. Legacy System Modernization Approaches And Strategies, URL: <https://www.entranceconsulting.com/legacy-system-modernization-approaches-and-strategies-entrance-consulting/>, (дата обращения: 24.03.2022).
28. Фаулер М. Рефакторинг. Улучшение существующего кода. С-пб., 2009.
29. Басов А.С. Методы рефакторинга кода // Вестник науки. 2020. №8 (29). URL: <https://cyberleninka.ru/article/n/metody-refaktoringa-koda> (дата обращения: 21.03.2022).
30. Ксензов М. Рефакторинг архитектуры программного обеспечения: выделение слоев // Труды ИСП РАН. 2004. №1. URL: <https://cyberleninka.ru/article/n/refaktoring-arhitektury-programmnogo-obespecheniya-vydelenie-sloev> (дата обращения: 21.03.2022).
31. IDC Top 10 Predictions For Worldwide IT, 2019 URL: <https://www.forbes.com/sites/louiscolombus/2018/11/04/idc-top-10-predictions-for-worldwide-it-2019/?sh=82f41db7b962>, (дата обращения: 21.03.2022).
32. Brent Frye 8 Steps for Migrating Existing Applications to Microservices, URL: <https://insights.sei.cmu.edu/blog/8-steps-for-migrating-existing-applications-to-microservices/>, (дата обращения: 22.03.2022).
33. Радостев Д.К., Никитина Е.Ю. Стратегия миграции программного кода из монолитной архитектуры в микросервисы // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2021. №2 (53). URL: <https://cyberleninka.ru/article/n/strategiya-migratsii-programmnogo-koda-iz-monolitnoy-arhitektury-v-mikroservisy> (дата обращения: 31.03.2022).
34. Балес А.И. Унифицированная модель данных и её применение в микросервисной архитектуре / DOI 10.25559/SITTO.16.202002.416-425 // Современные информационные технологии и ИТ-образование. – 2020. – Т. 16, № 2. – С. 416-425.
35. Слинкин Д.А. Современные подходы к модернизации веб-ресурсов образовательной организации // Вестник Шадринского государственного педагогического университета. 2019. №2 (42). URL: <https://cyberleninka.ru/article/n/sovremennye-podhody-k-modernizatsii-veb-resursov-obrazovatelnoi-organizatsii> (дата обращения: 22.03.2022).
36. Flavian, Carlos & Gurrea, Raquel & Orús, Carlos. (2009). Web design: A key factor for the website success. J. Systems and IT. 11. 168-184. 10.1108/13287260910955129.
37. Website Redesign Strategy – How to Redesign a Website in 2022, URL: <https://uxhacks.com/website-redesign/>, (дата обращения: 21.03.2022).

© В. М. Зарипова, И. Ю. Петрова

Ссылка для цитирования:

Зарипова В. М., Петрова И. Ю. Унаследованные информационные системы. проблемы и решения // Инженерно-строительный вестник Прикаспия : научно-технический журнал / Астраханский государственный архитектурно-строительный университет. Астрахань : ГАОУ АО ВО «АГАСУ», 2022. № 2 (40). С. 130–136.