

КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

corpinfosys.ru

Научно-популярный сетевой журнал

Стаценко М.Н.

Реализация процесса учета договоров с использованием нон-код платформы HoneyCode и метода Agile Scrum (часть 1)

Дорофеева Н.Е.

Становление корпоративных информационных систем в России

Арнаутов А.А.

Реализация бизнес-процессов транспортировки с использованием нон-код платформы «Интеграл» (часть 2)

Петров С.В.

Стратегия поддержки продуктивного запуска ERP-системы

Степанов Д.Ю.

Импортозамещение в проектах внедрения ERP-систем

Выпуск №18
2022 год

Научно-популярный сетевой журнал «Корпоративные информационные системы» (Corporate Information Systems)

Рецензируемое научно-популярное сетевое издание corpinfosys.ru, публикует оригинальные авторские статьи по актуальным вопросам корпоративных информационных систем и технологий. Свидетельство регистрации СМИ ЭЛ № ФС 77-71053 от 13.09.2017. Периодичность издания - четыре выпуска в год. Издаётся с марта 2018 года.

Миссия журнала - объединение теории с практикой анализа, проектирования, разработки, тестирования и внедрения корпоративных информационных систем из-за образовавшегося разрыва между двумя важнейшими составляющими проекта. Тематика журнала не ограничена рассмотрением лишь одного класса систем, обсуждаются всевозможные информационные, экономические, биотехнические и кибернетические системы, действующие в масштабе корпорации.

Принимаются к публикации работы не только научных сотрудников, специализирующихся по тематике журнала, но и руководителей проектов, консультантов, разработчиков, ключевых и конечных пользователей, непосредственно участвующих в процессе реализации корпоративных информационных систем. Достоверность, актуальность и новизна являются обязательными атрибутами публикуемых статей.

Важнейшими принципами издания являются доступность и открытость. Регламентированная и быстрая процедура рецензирования, а также электронный формат печати статей обеспечивают реализацию принципа доступности. Использование механизмов потока работ и операций в процессе обработки заявок на публикацию и возможность отслеживания статуса публикации задают принцип открытости. Рубрики журнала:

- теоретические основы корпоративных информационных систем;
- функционал корпоративных информационных систем;
- обзор современных корпоративных информационных систем;
- особенности реализации корпоративных информационных систем;
- интеграция корпоративных информационных систем;
- управление проектом внедрения корпоративных информационных систем;
- опыт внедрения корпоративных информационных систем.

Здравствуйте, уважаемые читатели!

Если вы знакомитесь с этим материалом, значит тематика корпоративных информационных систем Вам так же интересна, как авторам статей и членам редакционной комиссии.

Восемнадцатый выпуск журнала содержит подборку статей касающихся реализации процесса учета договоров с использованием нон-код платформы HoneyCode и метода Agile Scrum (часть 1), становлению корпоративных информационных систем в России, реализации бизнес-процессов транспортировки с использованием нон-код платформы «Интеграл» (часть 2), стратегии поддержки продуктивного запуска ERP-системы, импортозамещения в проектах внедрения ERP-систем.

Целью нашего журнала являются актуальность, доступность и открытость информации по тематике корпоративных информационных систем. Выражаю благодарность всем тем, кто способствует и помогает в подготовке выхода очередного выпуска.

Желаю Вам успешных проектов внедрения корпоративных информационных систем и наличия времени для обмена опытом на страницах нашего издания.

С наилучшими пожеланиями,
главный редактор журнала,
к.т.н., доц. РТУ МИРЭА
Степанов Дмитрий Юрьевич



Теоретические основы корпоративных информационных систем

Стаценко М.Н.

Реализация процесса учета договоров с использованием нон-код платформы HoneyCode и метода Agile Scrum (часть 1).....1

Дорофеева Н.Е.

Становление корпоративных информационных систем в России11

Арнаутов А.А.

Реализация бизнес-процессов транспортировки с использованием нон-код платформы «Интеграл» (часть 2).....19

Петров С.В.

Стратегия поддержки продуктивного запуска ERP-системы.....30

Степанов Д.Ю.

Импортозамещение в проектах внедрения ERP-систем.....36

Реализация процесса учета договоров с использованием нон-код платформы HoneyCode и метода Agile Scrum (часть 1)

Стаценко Михаил Николаевич

Аннотация: в статье автоматизируется процесс учета договоров в общеобразовательной организации на базе нон-код платформы. Выполнен анализ нон-код платформ и выбрано решение HoneyCode. Реализация проекта ведется с использованием метода Agile Scrum. Описываются детали метода Scrum. Далее идентифицируются и группируются в темы пользовательские истории, формируется и оценивается сложность бэклога продукта, создаются и планируются бэклоги спринтов.

Введение

В настоящее время в России наблюдается рост заинтересованности в применении ERP-систем для управления и, в частности, автоматизации бизнес-процессов предприятием [1]. В большинстве случаев компании малого и среднего бизнеса не могут позволить полномасштабное внедрение таких систем из-за большой продолжительности внедрения и как следствие его дороговизны [2]. Одним из вариантов решения подобного вопроса видится разработка собственного программного обеспечения для критически важных, профильных процессов силами штатных сотрудников. Для снижения затрат автоматизации можно использовать платформы по созданию приложений «без кода», ориентированные на специалистов без знаний языков программирования и опыта разработки.

С момента появления первых нон-код платформ, датирующихся серединой 90-х, многое изменилось: их возможности сильно увеличились, что это связано с изменениями ИТ-рынка, а также высокой стоимостью их работ. Существует потребность в большом количестве высококвалифицированных специалистов, которых не всегда хватает, поэтому разработчики начали развивать платформы для автоматизации работы самих программистов. Цель данной статьи состоит в разработке приложения по учёту договоров с использованием нон-код платформ на основе метода Agile Scrum.

1. Agile Scrum в проектах разработки программного обеспечения

1.1. Описание метода Agile Scrum

Метод Scrum является одним из самых популярных способов управления проектами по созданию программного обеспечения (далее - ПО). Scrum входит в семейство методов гибкой разработки Agile. Он представляет собой способ реализации и внедрения программного обеспечения на основе итерационной модели. Методология и сам метод базируется на соблюдении основных ценностей и принципов, описанных в манифесте Agile [3-4].

Действующими лицами в Scrum-проекте являются профильные технические специалисты, участвующие в нём от первой встречи и до полного завершения проекта. Scrum-команда, в центре которой находится владелец продукта, состоит из аналитиков, программистов, тестировщиков, администратора и скрам-мастера. Владелец продукта ведёт коммуникацию с заказчиком и собирает бизнес-требования, соединяет команду исполнителей с заказчиком и следит за ходом проекта.

Наиболее важным элементом Scrum является бэклог продукта, включающий собранные владельцем продукта пользовательские истории на основе информации, полученной от заинтересованных лиц заказчика, ранжированной по приоритету и сложности реализации. По завершении формирования бэклога продукта ведётся планирование спринтов, а именно: выбор задач, включаемых в бэклог спринта на основе оценки размера (в пунктах или идеальных человеко-днях), группировка в темы, определение длительности и скорости выполнения работ. Приоритет выполнения задач ведётся на основе комментариев заказчика, бизнес целесообразности и преимущества реализации базовых функций [5].

Стоит отметить, что в процессе реализации проекта допустимо вносить изменения в бэклог продукта. Обычно бэклог продукта создается на начальных этапах проекта, а бэклог спринта формируется командой на фазе планирования нового спринта. Поэтому, первый бэклог существует на протяжении всей разработки продукта, а бэклог спринта - только в рамках текущего спринта. По завершении каждого спринта производится презентация работающего продукта заказчику, а также обзор и ретроспектива, что позволяет оценить эффективность команды за время итерации, спрогнозировать ожидаемую скорость и выявить имеющиеся проблемы (рисунок 1.1).

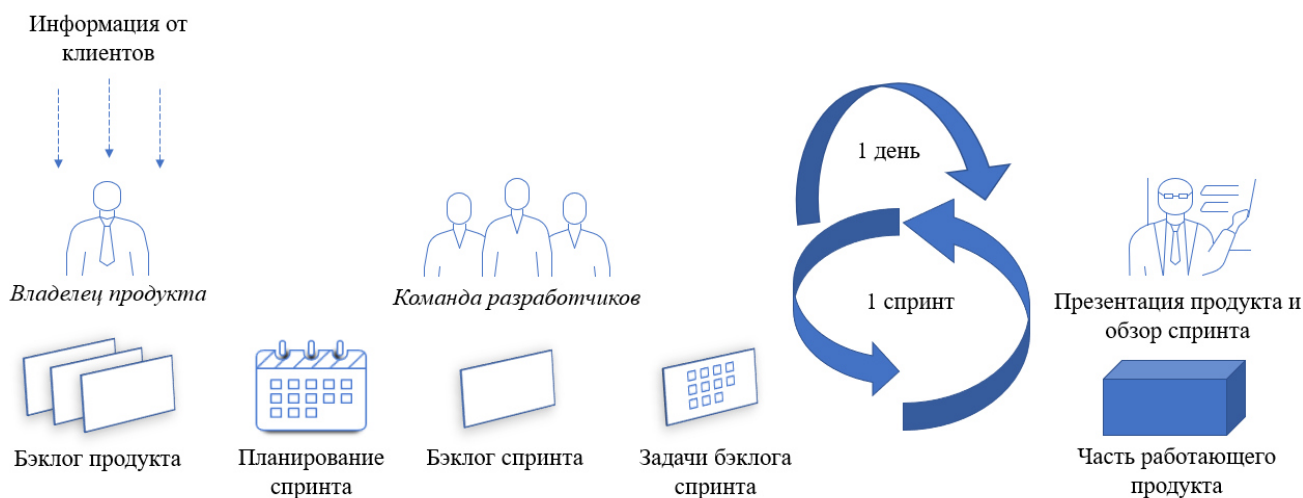


Рис. 1.1 - Схема реализации проекта Scrum

Scrum и его принципы позволяют достичь максимальной гибкости, прозрачности, простоты и отслеживаемости проделанной работы, то есть произвести переоценку приоритетов и внедрить новые функции или отказаться от них [6]. Верхнеуровневая последовательность действий для выполнения Scrum-проекта будет включать следующие шаги:

- собрать информацию от заказчика;
- сформировать бэклог продукта;
- выполнить приоритизацию бэклога;
- разбить бэклог на спринты;
- реализовать спринт;
- презентовать работающую часть приложения заказчику
- провести встречу с целью оценки скорости выполнения работ (при необходимости выполнить переоценку спринтов);
- возвращаться на пятый этап до тех пор, пока не будет разработан продукт в полном объёме.

1.2. Анализ нон-код платформ

С момент появления первых нон-код платформ их возможности значительно продвинулись вперёд. Нон-код платформы позволяют пользователям без опыта программирования создавать веб-сайты, приложения и организовать информационное взаимодействие. Несмотря на то, что нон-код-инструменты не способны решить абсолютно все задачи разработки программных проектов, подбор и использование адек-

ватного набора подобных инструментов, без сомнения, упрощает и ускоряет решение некоторых программных задач [7]. Анализ существующих платформ для разработки «без кода» дан в таблице 1.1.

Таблица 1.1. - Сравнение популярных non-код платформ

Наименование платформы	Возможность применения платформы интеграции	Количество бесплатных записей	Стоимость корпоративного тарифа (долл./мес.)	Функциональность платформы для реализации функционала будущего приложения
Bubble	нет	до 2500	88	достаточна
Glide	нет	до 1000	90	недостаточно
Webflow	нет	до 2500	73	достаточна
Adalo	нет	до 3000	70	достаточна
Bravo Studio	есть	до 3500	100	достаточна
HoneyCode	есть	до 4500	70	достаточно
Thunkable	нет	до 3000	85	достаточна

2. Идентификация требований

В качестве объекта автоматизации выбрана организация, занимающаяся деятельностью по дополнительному профессиональному образованию. Компания принадлежит к предприятию малого бизнеса, 70% сотрудников составляют преподавательский состав, остальные - специалисты финансового, правового, административного отделов и отдела маркетинга, а также директор компании. Критичный для компании бизнес-процесс - это учёта договоров, входящий в блок управления договорами. Существуют два подхода для работы с договорами: централизованный и децентрализованный. Первый вариант подразумевает наличие специального структурного подразделения, которая занимается исключительно договорами, второй - предполагает самостоятельное ведение контрактов конкретным исполнителями. В рассматриваемой компании применяется второй подход, зафиксированный в договорном регламенте, где описаны зоны ответственности и зависимости мероприятий.

Процесс ведения договоров делится на три этапа: подготовительный, на котором заключается, продлевается, расторгается или согласуется контракт; исполнения договорных обязательств, подразумевающий уведомление ответственных сотрудников, а также этап учёта и хранения. Отличительными особенностями управления контрактами являются: чёткая, зафиксированная регламентом, последовательность действий между четырьмя группами исполнителей, наличие юридического статуса договора, оперативное уведомление и передача информации ответственным сотрудникам.

2.1. Формирование бэклога продукта

Для идентификации требований к разрабатываемому продукту владельцы продукта применяют определённые механизмы сбора потребностей - предоставление в письменной форме описания того, чего требуется пользователю в виде: «Я как **должность/роль** хочу **требование** потому что **причина**». Подобный подход позволяет определить примерные характеристики продукта, поскольку в большинстве случаев при устном опросе конечные пользователи не могут описать конкретные функции продукта. В ходе рассмотрения специфики процесса управления договорами было выявлено четыре роли потенциальных пользователей будущего приложения: генеральный директор, менеджер по работе с клиентами, юрист и главный бухгалтер. По результатам окончания сбора требований, владелец продукта представил сформулированные пользовательские истории от заинтересованных отделов, представленных в таблице 2.1.

Скрам-команда дала оценку каждой истории в пунктах с использованием чисел Фибоначчи (1, 2, 3, 5, 8), где 8 - история, реализация, которая подразумевает наиболее массивный объём работ, 1 - история, не требующая больших затрат для реализации. Также истории были расставлены в приоритетном порядке в соответствии с бизнес-требованиями заказчика.

Таблица 2.1. - Оценка пользовательских историй

№	Пользовательские история	Оценка сложности (пункт)
1	Как генеральный директор я хочу иметь доступ к приложению с мобильных устройств (планшет, телефон) для ускорения процесса согласования и первичного ознакомления с данными договора	1
2	Как менеджер по работе с клиентами я хочу иметь доступ к приложению с мобильного устройства (планшета, телефона) с целью внесения	1

№	Пользовательские история	Оценка сложности (пункт)
	данных в любых условиях	
3	Как менеджер по работе с клиентами я хочу иметь возможность вводить информацию, соответствующую пунктам утвержденного стандартного печатного бланка договора для учёта договоров	8
4	Как менеджер по работе с клиентами я хочу иметь возможность удаления данных о договоре, контрагенте для отмены ошибочно внесённой информации	8
5	Как менеджер по работе с клиентами я хочу вносить информацию об ответственном за исполнение договорных обязательств с целью его оперативного оповещения	5
6	Как генеральный директор я хочу иметь возможность получать уведомления о статусе договоров для оперативного принятия решений и общей осведомленности	5
7	Как менеджер по работе с клиентами я хочу иметь возможность получать автоматические уведомления за неделю до окончания договора для проработки вопроса о продлении или конечном завершении контракта	5
8	Как генеральный директор я хочу, чтобы непосредственный исполнитель по договору получали уведомления о поступившем договоре и о статусе уже имеющегося	5
9	Как юрист я хочу оперативно получать уведомления о поступивших на меня договорах и о их статусе для понимания приоритета выполнения своих задач	5
10	Как менеджер по работе с клиентами я хочу иметь возможность применения фильтров и сортировки для навигации среди имеющихся договоров	5
11	Как менеджер по работе с клиентами я хочу иметь возможность поиска по уникальному значению для нахождения конкретного договора среди всех остальных	5
12	Как генеральный директор я хочу знать, какой сотрудник был определён ответственным за договор с целью осведомленности о занятости сотрудников	3
13	Как генеральный директор я хочу иметь возможность определять уровень доступа для сотрудников из соображения информационной без-	1

№	Пользовательские история	Оценка сложности (пункт)
	опасности	
14	Как генеральный директор я хочу импортировать имеющиеся записи в электронном формате (.CSV) для учёта прошлых договоров	1
15	Как менеджер по работе с клиентами я хочу иметь возможность прямого доступа к скан-копии физического договора для использования в работе	3

После окончания формирования бэклога продукта начинается этап планирования ближайших спринтов, их наполнение и определение продолжительности.

2.2. Формирование спринтов

В таблице 2.1 можно заметить несколько логических групп пользовательских историй, которые обычно называют темами. Темы позволяют легко ориентироваться в требованиях к будущему продукту, в таблице 2.2 представлены истории, сгруппированные в темы.

Таблица 2.2. - Группировка пользовательских историй по темам

История	Тема
Как генеральный директор я хочу иметь доступ к приложению с мобильных устройств (планшет, телефон) для ускорения процесса согласования и первичного ознакомления с данными договора	Вход с любого устройства, у которого есть доступ к интернету
Как менеджер по работе с клиентами я хочу иметь доступ к приложению с мобильного устройства (планшета, телефона) с целью внесения данных в любых условиях	
Как менеджер по работе с клиентами я хочу иметь возможность вводить информацию, соответствующую пунктам утвержденного стандартного печатного бланка договора организации для учёта договоров	Внесение и удаление информации
Как менеджер по работе с клиентами я хочу иметь возможность удаления данных о договоре, контрагенте для отмены ошибочно внесённой информации	

Как менеджер по работе с клиентами я хочу внести информацию об ответственном за исполнение договорных обязательств с целью его оперативного оповещения	
Как менеджер по работе с клиентами я хочу иметь возможность прямого доступа к скан-копии физического договора для использования в работе	
Как генеральный директор я хочу иметь возможность получать уведомления о статусе договоров для оперативного принятия решений и общей осведомленности	Отслеживание и получение статуса договора и информации об ответственном
Как генеральный директор я хочу, чтобы непосредственный исполнитель по договору получали уведомления о поступившем договоре или статусе уже имеющегося	
Как юрист я хочу оперативно получать уведомления о поступивших на меня договорах и о их статусе для понимания приоритета выполнения своих задач	
Как менеджер по работе с клиентами я хочу иметь возможность получать автоматические уведомления за неделю до окончания договора для проработки вопроса о продлении или конечном завершении контракта	
Как генеральный директор я хочу знать, какой сотрудник был определён ответственным за договор с целью осведомленности о занятости сотрудников	
Как менеджер по работе с клиентами я хочу иметь возможность применения фильтров и сортировки для навигации среди имеющихся договоров	
Как менеджер по работе с клиентами я хочу иметь возможность поиска по уникальному значению для нахождения конкретного договора среди всех остальных	
Как генеральный директор я хочу иметь возможность определять уровень доступа для сотрудников из соображения информационной безопасности	Определение уровня доступа для сотрудников
Как генеральный директор я хочу импортировать имеющиеся записи в электронном формате (.CSV) для учёта прошлых договоров	Импорт данных

Бэклоги спринтов требуют детализации, поэтому требуется указать конкретные технические задачи по их реализации. Кроме того, скрам-команда должна определить временные затраты на их выполнение, измеряемые в часах (таблице 2.3).

Таблица 2.3. - Определение задач и времени реализации

Тема	Задачи	Время реализации (часы)
Вход с любого устройства, у которого есть доступ к интернету	Нахождение наиболее подходящего решения (выбор нон-код платформы)	5
	Изучение документации по использованию платформы	7
Внесение и удаление информации	Реализация базы данных, соответствующим экранам	20
	Реализация интерфейса экранов	13
Отслеживание и получение статуса договора и информации об ответственном	Настройка всех требующихся видов уведомлений	25
	Проведение функционального тестирования	7
Ориентации в списках	Создание и настройка полей для фильтрации	5
Определение уровня доступа для сотрудников	Сбор информации с целью заведение учётных записей пользователей	5
	Заведение учётных записей пользователей в приложении	3
	Настройка уровня доступа	3
Импорт данных	Приведение накопленных данных к единой форме и их импорт	5

На этапе формирования бэклога продукта было принято решение о выставлении наибольшего приоритета для тех историй, которые задают скелет продукта и несут основной функционал приложения. После всех преобразований последовательность спринтов изменилась и дана в таблице 2.4. Кроме того, было принято решение об

определении продолжительности спринта в одну неделю при пятидневной рабочей неделе и продолжительности рабочего дня в восемь часов.

Таблица 2.4. - Разбивка задач по спринтам

Спринт	Тема	Дополнительные задачи
I	Вход с любого устройства, у которого есть доступ к интернету	Проектирование процессов, данных и приложения в AS-IS и TO-BE моделях
II	Внесение и удаление информации	
III	Отслеживание и получение статуса договора и об ответственном	
IV	Ориентации в списках, определение уровня доступа для сотрудников, импорт данных	Нагрузочное тестирование

Как можно увидеть из таблицы 2.4 в первый спринт были добавлены такие задачи как: проектирование моделей AS-IS и TO-BE, построение архитектуры данных и структуры приложения. Последний четвертый спринт включает допзадачу по нагрузочному тестированию. Метод Scrum позволяет декомпозировать задачи и переносить на следующие спринты те активности, которые не помещаются в рамках одного спринта. Поэтому было принято решение о реализации задачи построения базы данных и пользовательского интерфейса для заключения договоров в первом спринте, а создание таблиц данных и формы для ввода информации карточки контрагента, адреса организации и удаления сведений было отнесено на второй спринт.

Литература

1. Scrum Glossary [Электронный ресурс] // <https://www.scrum.org/resources/scrum-glossary> (дата обращения 10.04.2021).
2. Свищунова Ю.В. Современная структура корпоративной информационной системы // Компьютерные технологии в моделировании, управлении и экономике. - 2021. - № 13. - С. 67-70.
3. Larman C., Basili V.R. Iterative and Incremental Development: A Brief History // Computer. - 2003. - vol. 36, №6. - p. 47-56.

4. основополагающие принципы Agile-манифеста [Электронный ресурс] // Agile Alliance - Режим доступа: <http://agilemanifesto.org/iso/ru/principles.html> (дата обращения 10.12.2021).
5. Степанов Д.Ю., Вельсовский А.В. Применение Agile Scrum в проектах SAP // Корпоративные информационные системы. - 2018. - №1. - С. 9-17.
6. Кон М. Agile: оценка и планирование проектов. - М.: Альпина Таблишер, 2019.
7. Магомадов В.С. Платформы low-code и no-code как способ сделать программирование более доступным для широкой общественности // Международный научно-исследовательский журнал. - 2021. - № 6-1(108). - С. 100-103.

Выходные данные статьи

Стаценко М.Н. Реализация процесса учета договоров с использованием нон-код платформы HoneyCode и метода Agile Scrum (часть 1) // Корпоративные информационные системы. - 2022. - №2 (18) - С. 1-11. - URL: <https://corpinfosys.ru/archive/issue-18/194-2022-18-noncodehoneycode>.

Об авторе



Стаценко Михаил Николаевич - выпускник кафедры корпоративных информационных систем института информационных технологий РТУ МИР-ЭА. Тема выпускной квалификационной работы магистра «Разработка приложения по учету договоров с использованием нон-код платформ на основе метода Agile Scrum». Электронная почта: mail@corpinfosys.ru.

Становление корпоративных информационных систем в России

Дорофеева Наталья Евгеньевна

Аннотация: в статье изложена историческая справка о развитии ИТ- отрасли в нашей стране в период СССР в последние годы его существования. Кратко рассмотрена история становления и развития рынка автоматизированных корпоративных информационных систем в РФ с начала 90-х годов XX века по настоящее время.

История становления корпоративных информационных систем в России

Сегодня трудно представить работу современного предприятия без автоматизации процессов его деятельности, реализующих различные производственные и управленческие функции. В России начало внедрения основ автоматизации отдельных участков работы предприятия уходит в эпоху СССР.

В Советском Союзе в 1948 г. проблемы развития вычислительной техники становятся общегосударственной задачей. Были развернуты работы по созданию серийных электронно-вычислительных машин ЭВМ первого поколения (термин ЭВМ вместо «компьютер» употреблялся вплоть до конца 1980-х гг.).

Первыми изобретателями компьютера в СССР являются И.С. Брук и Б.И. Рамеев, совместно разработавшие проект цифровой ЭВМ с жестким программным управлением. В декабре 1948 г. они получили авторское свидетельство на изобретение «Автоматической цифровой машины».

«МЭСМ» (малая электронная счетная машина) – первый компьютер в СССР и в целом в континентальной Европе, была создана в 1951 г. под руководством С.А. Лебедева. Под его же руководством в дальнейшем были разработаны и сконструированы машины серии «М» (М-1, М-2 и другие ее модификации), из которых М-20 в 1960-х гг. была признана в СССР лучшей из отечественных машин. Специализированные ЭВМ, созданные под руководством С.А. Лебедева для системы противоракетной обороны, стали основой достижения стратегического паритета между СССР и США во время холодной войны.

В 1950 г. в Институте точной механики и вычислительной техники (ИТМ и ВТ) был организован отдел цифровых ЭВМ для их разработки и создания. В 1951 г. там

была спроектирована машина БЭСМ (большая электронная счетная машина), а уже в 1952 г. началась ее опытная эксплуатация.

В это же время конструкторское бюро (СКБ-245) Министерства машиностроения и приборостроения СССР во главе с М.А. Лесечко приступило к проектированию еще одного отечественного компьютера, получившего название «Стрела».

В 1953 г. в СССР начали серийное производство машин БЭСМ-1 и «Стрела», на базе которых в стране стали создаваться и работать вычислительные центры.

Производством ЭВМ и ЭВТ занимались следующие предприятия:

- ВНИИ Электромеханики (ВНИИЭМ, Москва);
- Институт электронных управляющих машин (ИНЭУМ, Москва);
- Научно-исследовательский центр электронной вычислительной техники (НИЦЭВТ, Москва);
- НИИ «Аргон» (СКБ-245);
- научно-исследовательский институт вычислительных комплексов (НИИВК, Москва);
- научно-исследовательский институт электронных вычислительных машин (НИИЭВМ, Минск);
- НПО «Научный центр» (Зеленоград) и многие другие.

Ассортимент выпускаемой ими продукции был значительным, среди них можно выделить, например, следующие (производились до 1993 г.):

- ЕС ТЭВМ;
- СМ;
- М;
- Оборудование «Электроника» разных видов и модификаций;
- ЭВМ Искра;
- Дельта-С и другие [1].

С появлением в 1950-х гг. цифровых программно-управляемых машин, родилась новая область прикладной математики: программирование. В то время над проблемами в области программирования одновременно работали советские и западные ученые-математики, а также специалисты технических направлений. Первоначально программы компилировались вручную на машинных языках (в машинных кодах). Программы были громоздкими, и отладка их была очень трудоемкой. Для упрощения приемов и методов компиляции и отладки программ была создана мнемотехника, структура которой близка к машинному языку, использующему символьную адреса-

цию. Ассемблеры, переводящие программы, написанные в мемокоде, на машинный язык и их расширенные макросы, используются по настоящее время [2].

В 1953 г. советский ученый А.А. Ляпунов предложил операторный метод программирования, заключающийся в автоматизации программирования, а алгоритм решения задачи был представлен в виде набора операторов, образующих логическую схему задачи. Ему приписывают формирование в нашей стране взгляда на программирование как на научную дисциплину. В числе первых советских ученых, занимавшихся разработкой технологий программирования, следует назвать также А.Т. Ершова, М.Р. Шура-Бура, В.М. Брябрина и др.

Первый софт (программное обеспечение, ПО) в современном понимании этого термина, появился в СССР в закрытых НИИ. Рынка программного обеспечения в то время просто не существовало и все ПО было узкопрофильным, предназначенным для определенных отраслей науки и промышленности. Разработка программного обеспечения в 1950-70 гг. хорошо финансировалась, поддерживая стремительное развитие советской науки. В это время проводились первые разработки в сфере информатики и кибернетики, создавались первые языки программирования, которые во многом и послужили основой для создания современного российского программного обеспечения [3].

Отечественные программисты в основном были математиками с университетским образованием. Качественный уровень их базисной подготовки позволил им занять лидирующие позиции в IT-мире в части модернизации и совершенствования языков программирования, программных систем, парадигм программирования, анти-вирусных программ и других направлениях. Российские программисты в своей деятельности по разработке отечественного софта и сегодня руководствуются интеллектуальным наследием советских и ведущих российских разработчиков ПО.

Завершение периода развития IT-отрасли в СССР приходится на начало и середину 1990-х гг.: время распада СССР, когда большинство советских IT-разработок было прекращено или значительно сокращено.

Корпоративные автоматизированные информационные системы как таковые свое становление получили в России в 90-е гг. XX в., в период начала в стране эпохи либеральных экономических отношений. В это время (1989-1990 гг.) в стране зарождались рыночные процессы в экономике и в обществе, появились кооперативы и первые

персональные компьютеры. А история развития IT-отрасли одновременно закончилась для СССР и началась снова уже для России.

Главная проблема отечественной IT-отрасли в этот период состояла в том, что в большинстве своем ее специалисты были первым поколением профессионалов российских IT-компаний, которые осваивали все методом собственных проб и ошибок. Изначально в РФ при разработке информационных систем для предприятий предпочтение отдавалось созданию программного обеспечения для автоматизации бухгалтерского учета и составления бухгалтерской/финансовой отчетности [4].

Тогда же на российском рынке появились и первые коммерческие разработки российских софтверных компаний, а также решения зарубежных компаний (Microsoft, IBM, Oracle, Sun), предназначенные для обеспечения комплексного управления ресурсами предприятий. С этого же времени такие системы стали называться информационными системами.

В настоящее время информационная система в деятельности компании рассматривается как программное обеспечение, реализующее её деловую стратегию и бизнес-процессы. Желательной целью является создание и развертывание единой корпоративной системы (КИС), удовлетворяющей информационные потребности всех сотрудников, служб и подразделений организации или компании. Российские софтверные компании в целом сегодня можно разделить на две категории: производители программного обеспечения, работающие на внутреннем рынке, и системные интеграторы, взаимодействующие как с российскими, так и иностранными вендорами. Компании внутреннего рынка преимущественно обслуживают крупные предприятия в платежеспособных отраслях. Есть еще ряд компаний, ориентированных на малый бизнес и отдельных покупателей.

Одним из первых и довольно успешных российских проектов по созданию программного обеспечения для автоматизации бухгалтерского учета является «1С: Бухгалтерия». История этой компании началась в 1988 г., создавалась она на базе НИИ Госкомстата. Институт занимался проектом в области баз данных. Основатель и генеральный директор компании - Борис Нуралиев. В 1992 г., продемонстрировав свой продукт «1С: Бухгалтерия», компании удалось заинтересовать налоговую службу и организовать их дальнейшее деловое сотрудничество. Такое взаимодействие помогало вовремя вносить необходимые обновления в программу с учетом изменений в законодательстве, в то время оно менялось особенно часто. Впоследствии и до настоящего времени «1С: Бухгалтерия» является одним из самых популярных решений для авто-

матизации бухгалтерского учета в России и СНГ. На данный момент количество решений, внедренных под названием «1С: Бухгалтерия» различных версий, насчитывает около полумиллиона предприятий [5].

Далее последовательно вышли на российский рынок корпоративных автоматизированных информационных систем со своими разработками отечественные фирмы, которые продолжают в настоящее время активную профессиональную деятельность, занимая значительную долю российского ИТ-рынка:

- корпорация «Галактика» - разработчик комплексных решений в области автоматизации управления производственно-хозяйственной и финансовой деятельностью предприятия. В 1986 г. были написаны первые заказные модули одноименной системы. На текущий момент развивает программные решения класса ERP, MES, EAM и HCM;
- СКБ «Контур» - один из первых разработчиков программного обеспечения в России, образованное в 1988 г. В начале 1990-х гг. занималась системами автоматического проектирования, позднее переключилась на автоматический учёт на предприятиях, разработав программы «Скат-S» и «Каскад»;
- корпорация «Тарус» - разработчик программных продуктов, предназначенных для автоматизации деятельности организаций сектора государственного и муниципального управления, а также коммерческих предприятий. В отечественной ИТ-отрасли компания работает с 1990 г.;
- фирма «Атлант-Информ» - разработчик интегрированной системы автоматизации управленческой деятельности предприятия (система «АККОРД»). Первые программные продукты вошли на российский рынок в 1994 г.;
- компания «БОСС-Корпорация» - разработчик программных продуктов по управлению финансово-хозяйственной деятельностью производственных, государственных и торговых предприятий и организаций. В ней автоматизированы разнообразные виды учета (бухгалтерский, оперативный и производственный), финансовое и производственное планирование, управление персоналом. Первые версии программные системы создавались, начиная с 1996 г. Наиболее известный программный продукт корпорации - решение по кадровому учету «БОСС-Кадровик»;
- компания «Монолит-Инфо» - российский разработчик корпоративных информационных систем класса ERP, один из ведущих поставщиков комплексных решений в области автоматизации управления производственно-хозяйственной и финансовой деятельностью предприятий, на базе собственной платформы «ERP Монолит», появившаяся на рынке еще в 1998 г.;

- компания «Рэйдикс» - один из ведущих российских разработчиков программного обеспечения для управления высокопроизводительными системами хранения данных. Компания вышла на российский рынок информационных технологий в 2009 г.;
- НТПЦ «Бизнес-Автоматика» - ведущий разработчик и поставщик IT-решений для органов государственной власти и крупных коммерческих организаций России, создан в 2010 г.

Помимо производителей российского программного обеспечения, сейчас ИТ-рынок содержит большое число системных интеграторов, занимающихся технологическим консалтингом и внедрением как отечественного, так и западного софта: Softline, IBS, Т1, КРОК, Айтеко, Диасофт и др. В связи с уходом с рынка большинства западных вендоров (SAP, Oracle, Microsoft, SAS и др.) в 2002 г., локальные российские интеграторы и производители корпоративных систем получили редкий шанс утвердиться в ИТ-области и вытеснить программные продукты международных корпораций. Вопрос импортозамещения программного обеспечения стал как никогда актуальным и всецело зависящим от нас самих.

Литература

1. Бусленко Н., Бусленко В. Беседы о поколениях ЭВМ. - М.: Молодая гвардия, 1977.
2. Брябрин В.М. Программное обеспечение персональных ЭВМ. - 3-е изд. - М.: Наука, 1990.
3. Очерки истории информатики в России/Ред. сост. Д.А. Пospelов, Я.И. Фет. Новосибирск: Науч. изд. центр ОИГГМ СО РАН, 1998.
4. История развития программирования в СССР и России. - URL: <https://www.evkoval.org/kursovye-raboty/istoriya-razvitiya-programmirovaniya-v-sssr-i-rossii> (дата обращения 10.01.2022).
5. Как строился ИТ-бизнес в России 90-х. - URL: <https://timeweb.com/ru/community/articles/it-v-rossii-90-h> (дата обращения 10.01.2022).

Выходные данные статьи

Дорофеева Н.Е. Становление корпоративных информационных систем в России // Корпоративные информационные системы. – 2022. – №2 (18) – С. 11-18. – URL: <https://corpinfosys.ru/archive/issue-18/195-2022-18-erpinrussia>.

Об авторе



Дорофеева Наталья Евгеньевна – к.ф.н., научный сотрудник ФГБОУ ВО «ЧГУ им. И.Н. Ульянова». Автор 13 научных статей по вопросам художественно-документальных жанров в английской литературе XVIII века. Электронный адрес: mail@corpinfosys.ru.

Реализация бизнес-процессов транспортировки с использованием non-код платформы «Интеграл» (часть 2)

Арнаутов Артемий Артурович

Аннотация: в статье для старта проектирования пользуются ранее сформированной картой процесса, которую детализируют до 4-го уровня декомпозиции. Это позволяет определить и спроектировать структуру баз данных, что также исключает необходимость детального моделирования процессов на низком уровне. В завершении дизайна создается макет программы в виде структуры приложения. Для спроектированных сущностей: процессы, данные и структура, выполнена программная реализация, для чего применялась non-код платформа «Интеграл».

2. Дизайн информационной системы

Типовая идея моделирования программы состоит в том, чтобы разбить ее на как минимум три составляющие: процессы, данные и непосредственно структура приложения [8-9]. Аналогичным образом поступают при построении ИТ-архитектуры в методологиях TOGAF, POSIX, методе Захмана и др. Последуем этому примеру и рассмотрим все требования через призму указанных составляющих.

2.1. Проектирование бизнес-процессов в моделях AS-IS и TO-BE

В ИТ кругах общеизвестным является тот факт, что организационная структура и бизнес-процессы предприятия, задающие бизнес-архитектуру, проектируются в двух базовых моделях: AS-IS и TO-BE. Для моделирования применяют CASE-средства, включающие в себя графические нотации как верхнего (ARIS VACD, IDEF0, BCM), так и нижнего (Cross WFD, BPMN SLD, UML AD, ARIS eEPC, IDE3 и DFD) уровней проектирования [10].

Теперь давайте разберемся, для чего же нужна AS-IS модель. Следуя модели зрелости компании, описывающей необходимость проектирования бизнес-процессов в зависимости от степени ее развития, обосновывается потребность в AS-IS схеме. Именно она позволяет спроектировать текущие подпроцессы и как следствие выявить «узкие места» в работе предприятия [11]. Цена вопроса – трудозатратное проектирование более чем 10 000 операций в выбранной графической нотации. Выполняя реинжиниринг процессов, TO-BE модель призвана исправить найденные недостатки. Одна-

ко, в случае внедрений крупных информационных систем типа ERP/ERP2 основной акцент TO-BE модель преимущественно делает на последовательное описание логики выполняемых операций одновременно с детальным разграничением ответственности сотрудников [12]. Следовательно, если принято решение о имплементации ERP-систем, потребность в AS-IS моделировании фактически отпадает.

Аналогичным образом проанализируем нотации моделирования бизнес-процессов в TO-BE модели. Разделение на верхнеуровневые и низкоуровневые графические схемы позволяет подойти к вопросу обсуждения процессов сверху вниз. Так как нотации проектирования на верхних уровнях не содержат деталей выполняемых операций по определению, их использование может быть легко заменено на подготовку карты процессов соответствующих уровней, пример которой дан в таблице 1.3. Схожим образом дело обстоит с низкоуровневыми нотациями: проектирование операций может быть замещено проработкой схемы процессов, в частности, если последовательность шагов и ответственность второстепенны.

Суммируя все вышесказанное, ограничимся в рамках проектирования лишь доработкой карты бизнес-процессов модели TO-BE. Но и здесь можно сократить объем выполняемых работ: карты процессов AS-IS и TO-BE практически совпадают для операций 1-3 уровней, где степень детализации минимальна. Проработка схемы процессов в TO-BE потребуется лишь для уровней 4-е и ниже, хотя на практике даже 3-го уровня детализации бывает достаточно. А чем, собственно, нам поможет карта процессов 4-го уровня? В-первую очередь она используется для проверки того, все ли операции реализованы в программе, во-вторую – подготовки тестовых сценариев приемочных испытаний, и, наконец, – управления изменениями в виде обучения сотрудников.

В ходе проработки бизнес-потребностей операции, отраженные на 3-м уровне карты процессов из таблицы 1.3, были детализированы. Это позволило зафиксировать подпроцессы 4-го уровня, максимально приближенные к исходным требованиям (таблица 1.1.). Результаты обогащения карты процессов на 4-м уровне даны в таблице ниже.

Таблица 2.1. Фрагмент карты процессов в модели TO-BE

№	Процесс 3-го уровня	№	Процесс 4-го уровня
1.2.1	Обработка потребностей в перевозке	1.2.1.1	Ведение карточки клиента

№	Процесс 3-го уровня	№	Процесс 4-го уровня
1.2.1	Обработка потребностей в перевозке	1.2.1.2	Создание/изменение/удаление заявки на транспортировку
1.2.1	Обработка потребностей в перевозке	1.2.1.3	Согласование заявки на транспортировку
1.2.1	Обработка потребностей в перевозке	1.2.1.4	Просмотр заявки на транспортировку
1.2.1	Обработка потребностей в перевозке	1.2.1.5	Преобразование заявки в транспортировку
1.2.1	Обработка потребностей в перевозке	1.2.1.6	Массовое преобразование заявки в транспортировку
1.2.1	Обработка потребностей в перевозке	1.2.1.7	Формирование отчетности по заявкам на транспортировку
1.2.2	Управление перевозками	1.2.2.1	Ведение карточки перевозчика
1.2.2	Управление перевозками	1.2.2.2	Ведение маршрутов
1.2.2	Управление перевозками	1.2.2.3	Создание/изменение/удаление транспортировки
1.2.2	Управление перевозками	1.2.2.4	Печать сопроводительных документов
1.2.2	Управление перевозками	1.2.2.5	Просмотр транспортировки
1.2.2	Управление перевозками	1.2.2.6	Расчет фактических затрат транспортировки
1.2.2	Управление перевозками	1.2.2.7	Формирование отчетности по транспортировкам

2.2. Моделирование структуры таблиц баз данных

Оптимальным с точки зрения построения структуры баз данных является максимальное использование результатов проектирования. Так, в таблице 2.1 даны операции 4-го уровня детализации. Они сформулированы достаточно четко, чтобы соответствовать идентифицированным требованиям, это позволяет практически однозначно определить объекты хранения данных. Предложим классы данных для проектируемого приложения, а далее уточним их и детализируем на таблицы баз данных, доведя до 3-й нормальной формы. Результаты проделанной работы можно найти в таблицах 2.2-2.3, где под ID подразумевается ключевое поле или составной ключ заданной таблицы.

Таблица 2.2. Классы данных

№	Процесс 4-го уровня	Класс данных
1.2.1.1	Ведение карточки клиента	Клиенты

№	Процесс 4-го уровня	Класс данных
1.2.1.2	Создание/изменение/удаление заявки на транспортировку	Заявка на транспортировку, Пользователи, Материалы
1.2.1.3	Согласование заявки на транспортировку	Заявка на транспортировку
1.2.1.4	Просмотр заявки на транспортировку	Заявка на транспортировку
1.2.1.5	Преобразование заявки в транспортировку	Транспортировка
1.2.1.6	Массовое преобразование заявки в транспортировку	Транспортировка
1.2.1.7	Формирование отчетности по заявкам на транспортировку	Транспортировка
1.2.2.1	Ведение карточки перевозчика	Перевозчики
1.2.2.2	Ведение маршрутов	Маршрут
1.2.2.3	Создание/изменение/удаление транспортировки	Транспортировка
1.2.2.4	Печать сопроводительных документов	Транспортировка
1.2.2.5	Просмотр транспортировки	Транспортировка
1.2.2.6	Расчет фактических затрат транспортировки	Фрахт
1.2.2.7	Формирование отчетности по транспортировкам	Транспортировка

Таблица 2.3. Структура таблиц баз данных в 3-й НФ

Класс/таблица	Поле	Тип данных
Партнеры (клиенты и перевозчики)	ИНН	Числовой
	КППП	Числовой
	Название краткое	Текстовый
	Название юридическое	Текстовый
	Адрес юридический	Текстовый
	Адрес отгрузки	Текстовый
	ID (ИНН, Краткое название)	Текстовый
Заявка на транспортировку (заголовок)	№ заявки	Числовой
	Партнер-клиент (ID)	Текстовый
	Адрес погрузки	Текстовый
	Дата погрузки	Дата
	Время погрузки	Время
	Дата отгрузки	Дата
	Время отгрузки	Время
	Вес	Числовой

Класс/таблица	Поле	Тип данных
	Пользователь (ID)	Текстовый
	ID (№ заявки)	Числовой
Заявка на транспортировку (позиция)	№ заявки	Числовой
	№ позиции	Числовой
	Материалы (ID)	Числовой
	Вес	Числовой
	ID (№ заявки, № позиции)	Числовой
Материалы	№ материала	Числовой
	Название	Текстовый
	Единица измерения	Текстовый
	Вес	Числовой
	ID (№ материала)	Числовой
Пользователь	Фамилия	Текстовый
	Имя	Текстовый
	Отчество	Текстовый
	Пол	Текстовый
	Дата рождения	Дата
	E-mail	Текстовый
	ID (Фамилия, Имя, Отчество, дата рождения)	Текстовый
Транспортировка	№ транспортировки	Числовой
	Партнер-Перевозчик (ID)	Текстовый
	Вид транспорта	Текстовый
	Маршрут (ID)	Числовой
	Плановое расстояние (км.)	Числовой
	Плановая цена за единицу расстояния (руб.)	Числовой
	Плановая стоимость (руб.)	Числовой
	Заявка на транспортировку (заголовок) (ID)	Числовой
	ID (№ транспортировки)	Числовой
Маршрут	№ маршрута	Числовой
	Вид транспорта	Текстовый
	Адрес погрузки	Текстовый
	Адрес отгрузки	Текстовый
	Плановое расстояние (км.)	Числовой
	Время в пути (час)	Числовой
	ID (№ маршрута)	Числовой
Цены	№ записи цены	Числовой
	Вид транспорта	Текстовый
	Цена за единицу расстояния (руб.)	Числовой
	ID (№ записи цены)	Числовой
Фрахт	№ документа фрахта	Числовой
	Дата создания	Дата
	Фактическое расстояние (км.)	Числовой

Класс/таблица	Поле	Тип данных
	Фактическая стоимость (руб.)	Числовой
	Транспортировка (ID)	Числовой
	ID (№ документа фрахта)	Числовой

2.3. Формирование карты приложения

Структура данных, обозначенная в таблице 2.3, демонстрирует ключевые таблицы для ведения данных. На базе этих таблиц готовится карта приложения, описывающая представление будущей программы с точки зрения пользовательского интерфейса. Так как фактически все спроектированные данные можно разделить на три группы: основные, переменные и отчетные, поэтому представление главного меню программы возможно задать следующим образом (рис. 2).

Главное меню позволяет запустить ту или иную форму ввода: из области документов заводятся транзакционные данные, включающие заявки, транспортировки и фрахты; область основных данных обеспечивает ведение справочников партнеров, маршрутов и цен; нижестоящая секция позволяет запускать отчетность с различными аналитическими признаками. Особенность разрабатываемого приложения состоит в том, что:

- создание документов транспортировок предполагается делать со ссылкой на заявку, часть данных из которой передается в последующий документ;
- плановое расстояние в транспортировки рассчитывается на основании тройки параметров «вид транспорта-адрес погрузки-адрес отгрузки». Ожидаемая стоимость считается как произведение предполагаемого расстояния на значение, найденное из двойки «вид транспорта-цена единицы расстояния»;
- фактические затраты уточняются в документе фрахта, формируемом по результатам завершения доставки.

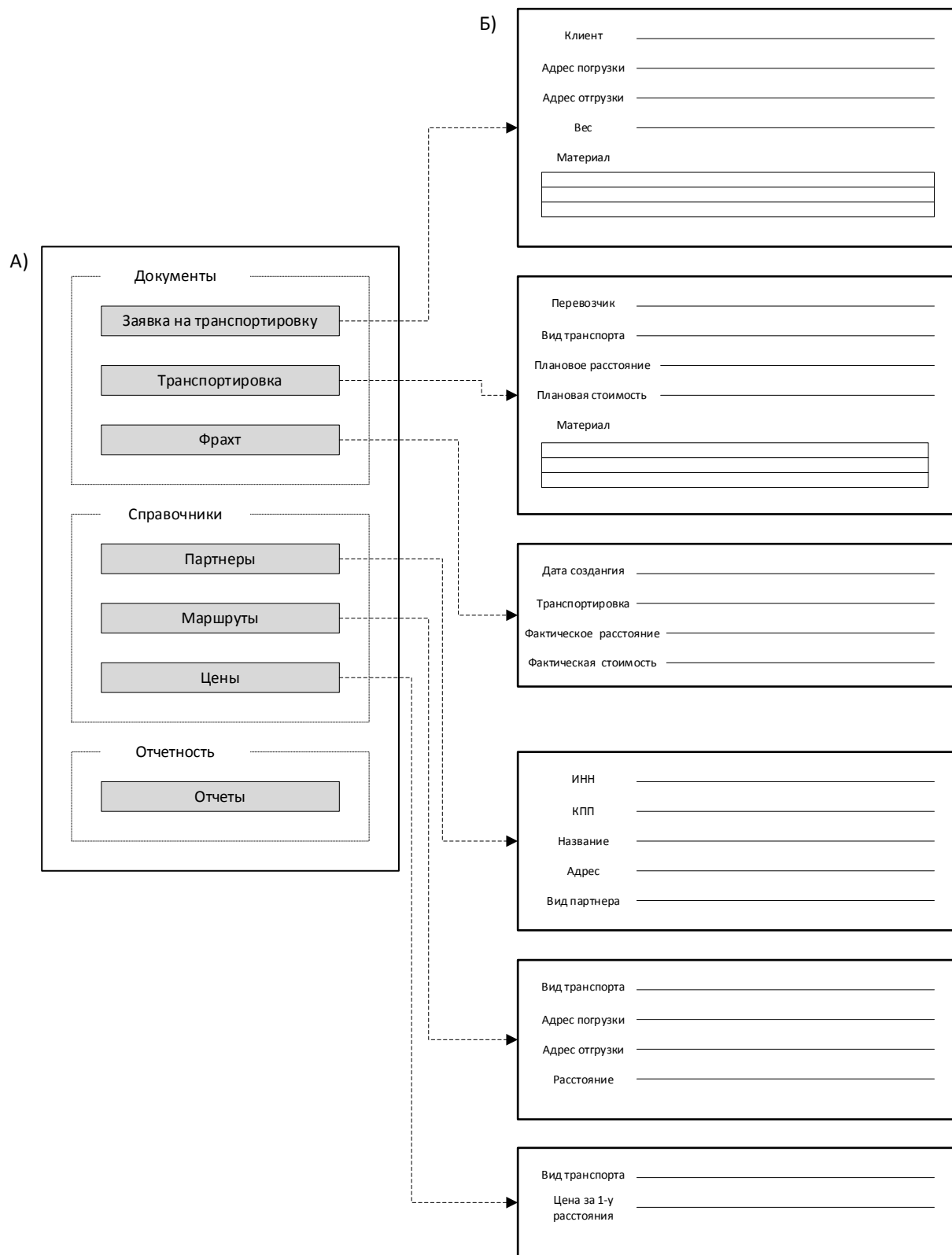


Рис. 2. Структура разрабатываемого приложения:
а) главное меню; б) формы ввода данных

3. Разработка приложения

В качестве среды реализации приложения выбрана бесплатная non-код платформа «Интеграл», а также система управления базами данных MySQL, адаптированная под веб разработку. Последовательно велись настройка структуры данных согласно результатам раздела 2.2, а также построение форм ввода данных, описанных в пункте 2.3. Фрагменты полученных программных форм приложения продемонстрированы на рисунках ниже (рис. 3-5).

Рис. 3. Форма создания документов транспортировок

Адрес доставки	Время пути (часов)	Расстояние (км)	Время разгрузки
Красная площадь, 1	1.50	40	0.50
САМОВЫВОЗ	0.00		
Станционная 52	1.50		0.75
ул. Лакина	2.00		
Энергетик	1.50	24	

Рис. 4. Форма заведения данных маршрута

Заказ	Номер	Статус	Клиент	Продукт	Адрес	Кол-во (м3)	Сумма	1С	Пользователь	Оплата (сумма)	Примечание	Цена
12.08.2020	1	В работу	АО им Лакина	200-02		(2)		X		(0)	vdfdg 2222	
07.11.2020	1	В работу	АО им Лакина	100-С5	wwewewew	(2)		X		(0)		
07.11.2020		В работу	Частное лицо	200-02		(1)				(0)		
12.11.2020		В работу	Андрианов	350-04		(0)				(0)		
11.05.2021		В работу	АО им Лакина	200-06		(1)				(0)	vdfdg 2222	
27.05.2021		В работу	Артем	300-07	Ramenki 9-4	(2)				(0)		
28.05.2021	1	В работу	Артем	400-03	Красная площадь, 1	(1)		X		(0)	ФСО пропустит 89108880000	5600.00

Рис. 5. Пример отчета по созданным документам транспортировок

Заключение

Работа посвящена использованию конфигурируемых программных платформ, не требующих программирования для реализации приложения. В качестве примера выбрана бесплатная non-код платформа «Интеграл», ориентированная на интернет программы. Реализация приложения для автоматизации бизнес-процессов перевозок потребовала на начальных этапах работы идентификации и приоритизации требований, для чего предварительно готовилась карта процессов предприятия. Карта процессов предприятия, детализированная сначала до 3-го уровня декомпозиции, а затем и до 4-го, позволила ускорить сбор требований, определить автоматизируемые процессы, а также уточнить те объекты данных, которые в дальнейшем подлежали проектированию. Классы данных помогли быстро уточнить и смоделировать структуру баз данных, реализованных на следующем этапе работ в MySQL.

Построение карты процессов, проектирование структуры таблиц, а также моделирование предполагаемых форм ввода данных разрабатываемого приложения позволили разработать его в срок, изначально определенный до старта работ. Ключевая особенность текущей статьи состоит в том, что здесь специально исключен этап проектирования бизнес-процессов, что в большей части литературных источников описано как обязательный шаг. Вместо этого максимально применяется карта процессов. В действительности можно было попробовать сузить область проектирования процессов только моделированием операций на последних уровнях декомпозиции, однако это было не столь критично в виду сроков проекта.

Завершая данную работу, хочется еще раз отметить, что классический подход к реализации проекта согласно каскадной методологии внедрения программного обсе-

чения, требует проработки трех важных блоков: процессы, данные и структура приложений, что обычно делается в документе проектного решения. В ходе проработки данного материала стало очевидным, что важно не столько описание процессов на всех уровнях детализации в TO-BE, сколько наличие карты процессов, а также лишь части смоделированных операций, в частности на 4-м уровне детализации.

Литература

1. Volition L.X. No-code 101: 270 tools to build websites, apps and software without writing a single line of code. Seven marage. 2022. 301 p.
2. Documents Associated with Business Process Model and Notation Version 2.0 [Электронный ресурс] // Сайт Object Management Group. – Режим доступа: <http://www.omg.org/spec/BPMN/2.0/> (дата обращения 10.01.2022).
3. Логистика / Дыбская В.В. и др. – М.: Эксмо, 2009. – 944 с.
4. Новоковский Е.А., Степанов Д.Ю., Шутихина Ю.В. Особенности ведения транспортировок в SAP ERP // Корпоративные информационные системы. – 2019. – №2(6). – С. 39-57. – URL: <https://corpinfosys.ru/archive/issue-6/58-2019-6-transport>.
5. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем: учебное пособие. – Ростов н/Д.: Феникс, 2009. – 508 с.
6. О'Лири Д. ERP системы. Современное планирование и управление ресурсами предприятия. Выбор, внедрение и эксплуатация / Пер. с англ. Водянова Ю.И. – М.: Вершина, 2004. – 272 с.
7. Степанов Д.Ю. Подготовка функциональных спецификаций для разработки корпоративных информационных систем на примере SAP ERP (часть 1) // Корпоративные информационные системы. – 2019. – №3(7). – С. 29-52. – URL: <https://corpinfosys.ru/archive/2019-7/66-2019-7-functionalspec>.
8. Петрова Е. А., Фокина Е. А. Информационный менеджмент. – М.: Лань, 2020. – 215 с.
9. Остроух А.В., Суркова Н.Е. Проектирование информационных систем. М.: Лань, 2019. 164 с.
10. Ковалев С., Ковалев В. Секреты успешных предприятий: бизнес-процессы и организационная структура. – М.: БИТЕК, 2012. – 498 с.

11. Лодон Дж., Лодон К. Управление информационными системами. / Пер. с англ. под ред. Трутнева Д.Р. – СПб.: Питер, 2005. – 912 с.
12. Олейник П.П. Корпоративные информационные системы: учебник для вузов. – СПб.: Питер, 2012. – 175 с.

Выходные данные статьи

Арнаутов А.А. Реализация бизнес-процессов транспортировки с использованием нон-код платформы «Интеграл» (часть 2) // Корпоративные информационные системы. – 2022. – №2 (18) – С. 19-29. – URL: <https://corpinfosys.ru/archive/issue-18/198-2022-18-noncodeintegral>.

Об авторе



Арнаутов Артемий Артурович – выпускник кафедры корпоративных информационных систем института информационных технологий РТУ МИРЭА. Тема выпускной квалификационной работы магистра «Реализация ключевых бизнес-процессов логистики транспортировки с использованием нон-код платформы». Электронная почта: mail@corpinfosys.ru.

Стратегия поддержки продуктивного запуска ERP-системы

Петров Сергей Владимирович

Аннотация: в статье дается описание того, что понимание и заблаговременная проработка активностей, необходимых для поддержки ERP-системы, обеспечивает безболезненный этап ее продуктивного запуска. Пользователям необходимо знать, к кому обращаться за помощью в случае возникновения вопросов, дефекты должны маршрутизироваться и разрешаться, а для обработки новых требований важно наличие специального регламента. Все это является содержанием стратегии поддержки продуктивной эксплуатации системы ERP.

Введение

Продуктивный запуск является ключевым событием проекта внедрения ERP-систем. Обычно ему предшествуют весьма трудозатратные мероприятия продуктивной миграции, а также технического и бизнес катеров. Поддерживающими продуктивный запуск документами являются: план Б, план обеспечения непрерывности бизнеса, план отказа и стратегия поддержки [1]. Каждый из документов предназначен для решения своей специфичной задачи, но все они служат лишь одному: максимально безболезненно запустить программное продуктивное решение и заставить пользователей в нем работать на регулярной основе.

Поддержка продуктивного запуска пользователей или, как часто ее называют, гиперопека, преимущественно длится 4-е недели, после чего программное решение передается на сопровождение ИТ-службе заказчика. Обычно 4-х недель продуктивной поддержки хватает для помощи заказчику в закрытии одного бухгалтерского и налогового периодов в новой ERP-системе, однако бывает случаи, когда ее продлевают на более длительный период [2]. Сложность и продолжительность гиперопеки зависит также от того, имеется ли на стороне заказчика специально выделенная и организованная ИТ-служба поддержки.

Цель данной статьи представляется в рассмотрении особенностей поддержки продуктивного запуска ERP-системы, что должно обеспечить более «мягкий» ее запуск и слаженный механизм реагирования и исправления программных инцидентов.

1. Стратегия поддержки

Для начала давайте определимся, что же такое стратегия и какие задачи она призвана решать. Стратегия поддержки характеризует один из подходов к решению проектной задачи, что применительно к продуктивному запуску есть обеспечение своевременного и качественного устранения программных ошибок, а также передача ответственности за систему пользователям. В контексте стратегии должно быть описано, как указанные сложности будут разрешаться.

Начнем с разрешения дефектов, возникающих при продуктивном запуске. Чаще всего на стороне заказчика организуется команда поддержки, которая в режиме реального времени способна собирать, маршрутизировать и разрешать ошибки ERP-системы с вовлечением проектной команды. В свою очередь время реагирования на ошибки системы различается для гиперопеки продуктивного запуска и поддержки регулярных процессов [3]. Для формализации порядка работы службы поддержки вводят термин линия поддержки. На практике выделяют 4-е линии поддержки: регистраторы дефектов, маршрутизаторы, эксперты и команда вендора:

- первая линия поддержки решает задачи регистрации обращений от пользователей и их заведения систему ITSM;
- распределение дефектов по ответственным функциональным командам для их анализа и решения осуществляется на второй линии;
- разрешение инцидентов силами экспертов ведется на третьей линии поддержки;
- в случае, если дефект представим недоработкой стандартного функционала ERP-системы, он передается на сторону вендора.

Если в договоре на внедрение явно указан период гарантийной поддержки, то при наступлении гарантийного случая к разрешению инцидентов на 3-м уровне могут привлекаться специалисты исполнителя. Когда имплементирует «самописная» информационная система исполнителя, исправление ошибок ее базового функционала на 4-го уровне будет также его ответственностью.

Допустим, вы организуете ИТ-службу, состоящую из 4-х линий поддержки, что дальше? Теперь нужно задать регламент их работы, приоритеты дефектов и время их обработки. Чем выше приоритет дефекта, тем быстрее он должен быть разрешен. Если служба ИТ-поддержки организуется «с нуля» в рамках проекта внедрения ERP-системы, требуется отретипировать ее работу. Для чего усеченный состав команды ИТ,

а также функционал технической инфраструктуры используются уже на этапе интеграционного теста программного решения.

Зарегистрированный дефект маршрутизируется на последующую линию поддержки. В случае отнесения дефекта к категории, характеризующей новое требование, еще не реализованное в системе, необходимо проработать порядок последующего управления изменениями (запросами на изменения, Change Request). В частности, для каждого запроса потребуется сформулировать бизнес обоснование, задать критичность для бизнеса, привести предварительный расчет трудозатрат и стоимости реализации. Для подобных запросов организуются дополнительные встречи с управляющим комитетом, на котором запрос рассматривается, защищается, принимается решение о его включении/невключении, объем текущей реализации.

Следующий важный момент, который следует упомянуть в стратегии поддержки, - вовлечение бизнес пользователей в разрешение дефектов. 1-я линия поддержки при обращении обычно ссылается на существующие инструкции и ключевых пользователей как носителей знаний. Регистрацию дефектов часто поручают делать исключительно ключевым, но не конечным пользователям. Тем самым знания постепенно накапливаются и генерируются внутри пользователей, т.е. владельцев ERP-системы.

В высококритичных проектах используют документы, поддерживающие запуск: плана Б и отката системы. Оба документа направлены на обработку исключительно неблагоприятных ситуаций, так первый подразумевает использование обходных путей в случае затягивания сроков разрешения критичных дефектов, а второй - полный отказ от недавно имплементированной ERP-системы, внедрение которой признали unsuccessful.

2. План Б и откат системы

Предположим, в процессе поддержки запуска был зарегистрирован инцидент с самым высоким приоритетом. Однако даже в этом случае время его исправления может затянуться на целый рабочий день в виду как технических, так и организационных сложностей. Для обработки подобных ситуаций прорабатывается план Б. Формирование плана начинается на этапе приемочного тестирования, где определяются наиболее критичные функции ERP-системы в разрезе бизнес направлений, которые могут дать сбой именно при запуске. Далее для каждой функции выбирается обходной путь, позволяющий достичь тех же результатов иным способом. Классическим примером служит формирование сопроводительных документов: если в ERP-системе невозмож-

но сформировать и распечатать форму ТОРГ-12, то обходным путем будет являться ее ручная подготовка вне системы в Excel-файле. Кроме того, прорабатываются и согласуются предпосылки для начала использования плана Б, так как его применение обязательно потребует дальнейших нестандартных действий в ERP-системе для отражения последующих корректировок. Обычно план Б готовится ключевыми пользователями, кто в последующем и будет его исполнять.

Крупномасштабное внедрение ERP-систем на предприятии обычно ведется через схему опытно-промышленной эксплуатации в рамках пилотного проекта. Так, целесообразность имплементации корпоративной информационной системы на всех подразделениях компании определяется по результатам ее использования на одном из тестовых предприятий. Если оказывается, что применение ERP-системы не дает ожидаемых результатов в тестовой организации, от нее отказываются, для чего применяется план отката. План отката описывает порядок шагов для отказа от использования нового ИТ-решения и перечень мероприятий для возобновления работы в «старой» системе. Если активности плана Б запускают после согласования с руководителем проекта, то план отката - в рамках обсуждения с управляющим комитетом, так как существенна финансовая сторона вопроса.

Заключение

Как мы видим, стратегия поддержки состоит в организации грамотной работы ИТ-команды, для обеспечения слаженного реагирования на возникающие недоработки внедряемой ERP-системы:

- пользователи должны знать, к кому и как обращаться за помощью в случае возникновения вопросов, что обычно решается через ключевых пользователей и ссылкой на инструкции;
- в случае регистрации дефектов, они должны четко маршрутизироваться и решаться, не важно, присваиваются ли они внутреннему эксперту или поставщику программного обеспечения;
- для обработки новых требований необходимо проработать порядок управления изменениями содержания, т.е. регламент работы с запросами на изменения;
- при наличии высокоприоритетного дефекта, блокирующего выполнение критичного бизнес-процесса компании, прибегают к плану Б, задающему обходной путь решения задачи;

- и, наконец, если ИТ-система не оправдывает изначальных ожиданий и не позволяет достигнуть целей проекта, осуществляется откат к использованию «старой» системы, что записано в плане отката.

По результатам завершения этапа гиперопеки, заказчик получает ERP-систему в постоянное пользование и сопровождение. Поэтому порядок работы ИТ-службы должен быть изменен: задается согласованный уровень обслуживания системы, известный как SLA (Service Level Agreement), уточняется время реагирования на дефекты, определяется продолжительность работы команды поддержки в неделю, например, 8x5 или 24x7, и др. В общем случае, регулярную ИТ-поддержку может вести как сам заказчик, так и нанятая им организация по отдельному договору аутсорсинга.

Несмотря на кажущуюся простоту поддержки запускаемого ERP-решения, процесс реагирования, взаимодействия и исполнения инцидентов должен быть четко продуман и регламентирован заранее. Организация ИТ-поддержки как в организационном, так и техническом смысле должна вестись уже на этапе тестирования ERP-системы. Подход к организации гиперопеки запускаемого решения и регулярной ее поддержки обычно отличаются с точки зрения регламентированности и оперативности устранения дефектов, что должно найти отражение в стратегии гиперопеки.

Литература

1. Петрова Е. А., Фокина Е. А. Информационный менеджмент. – М.: Лань, 2020. – 215 с.
2. Остроух А.В., Суркова Н.Е. Проектирование информационных систем. М.: Лань, 2019. 164 с.
3. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем: учебное пособие. – Ростов н/Д.: Феникс, 2009. – 508 с.

Выходные данные статьи

Петров С.В. Стратегия поддержки продуктивного запуска ERP-системы // Корпоративные информационные системы. – 2022. – №2 (18) – С. 30-35. – URL: <https://corpinfosys.ru/archive/issue-18/199-2022-18-erpgolivehypercare>.

Об авторе



Петров Сергей Владимирович - эксперт по разработке программных решений в банковской, торговой и производственной сферах. Специализируется на языках программирования высокого уровня C++, Java и Transact SQL. Имеет более чем 10-летний опыт разработки приложений. Принимал участие в проектах разработки аналитических, экспертных, биотехнических и корпоративных систем. Электронный адрес: mail@corpinfosys.ru.

Импортозамещение в проектах внедрения ERP-СИСТЕМ

Степанов Дмитрий Юрьевич

Аннотация: в статье обсуждается вопрос импортозависимости проектов внедрения ERP-систем. Информационные системы рассматриваются с точки зрения уровней внедрения: проект и изменения, а также процессы, данные, приложения и техника. Каждый их уровней имплементирования оценивается с точки зрения импортозависимости. Подчеркивается высокая зависимость ERP-проектов от импортного «железа», нежели самих внедряемых приложений.

Введение

Сегодняшняя ситуация вокруг спецоперации в Украине и последовавшие рестрикции со стороны европейских стран и США обнажили и без того известные проблемы импортозамещения западной продукции. Не исключением в этом вопросе стало программное обеспечение (далее - ПО), о старте импортозамещения которого заговорили еще в 2016 года. По изначальному плану замещение программного обеспечения должно было завершиться в 2024 году. Однако, не смотря на столь амбициозный план, ощутимых результатов это не дало как несколько лет назад, так и сейчас. Да, конечно, на протяжении всего этого времени публиковалось множество материалов на тему успешного замещения части западного ПО отечественным, однако это была лишь каплей в море ИТ-решений и технологий.

Последовавший в начале 2022 года уход из России практически всех известных вендоров от Microsoft, до SAP и Oracle, показал нашу сильную зависимость от прививаемого годами зарубежного ИТ-сервиса. По существу, начало полноценного замещения зарубежного ПО началось именно с этого момента, когда российские компании поставили перед фактом остановки поддержки, продления лицензий и работы самого ИТ-решения. Большинство ИТ-менеджеров и директоров судорожно стало думать о переходе на российские аналоги, в частности на продукты 1С, как базового корпоративного решения. Однако, использование нового ПО на предприятии возможно лишь после его имплементирования, процесс которого оказался также сильно завязан на страны запада.

В этой статье мы рассмотрим вопрос импортозамещения приложений класса ERP и ERP2, принимая во внимание тот факт, что это далеко не единственный компонент корпоративной информационной системы (далее - КИС). Для целей удобства повест-

ования термины программное обеспечение, информационные и корпоративные системы, а также ERP-системы для нас будут тождественны. В работах [1-2] выделяют четыре основных уровня задач для внедрения КИС, к которым относят процессы, данные, приложения и технику, а также два вспомогательных: управление проектом и изменениями. Собственно говоря, процесс внедрения и импортозамещения ERP-систем мы будем рассматривать сквозь указанные 6-ть параметров.

1. Импортозамещение в проектах внедрения ERP-систем

Начнем рассмотрение с уровня управления проектом внедрения КИС, где задается метод имплементирования. Классические модели внедрения: каскадная, итерационная и спиралевидная [3], никак не привязаны к конкретному программному обеспечению, импортозаместить которые мы и планируем в данной работе. Западные методологии внедрения, такие как: ASAP, SAP Activate, MDSS, OUM, Scrum, FDD, RAP и XP лишь уточняют классические модели имплементации, но не заменяют их. При желании можно найти их исчерпывающее описание в сети интернет. Существующее программное обеспечение для управления проектами Project от Microsoft, используется больше для построения критического пути и расчет сроков, нежели полноценного управления ИТ-проектом. Приложение весьма негибко ко всевозможным изменениям в проектах: сроки, ресурсы, затраты и др. Разрекламированный PMBoK [4], несмотря на частое упоминание в среде руководителей проектов, представляется не более чем сводом знаний, накопленным за последние несколько десятков лет, при желании легко может быть заменен ГОСТами. Кроме того, его содержание не есть тайна, все активности проекта относят к 11-ти параметрам, каждый из которых нужно обрабатывать на ежедневной основе в ходе управления проектом. Поэтому ни первое, ни второе, ни третье никак не блокирует внедрение российских программных систем.

Теперь проанализируем уровень управления изменениями. По определению данный уровень необходим для того, чтобы изменения, в частности обусловленные внедрением ERP-системы, прижились в повседневной работе компании. Что обычно делается путем:

- вовлечения ключевых пользователей в проект внедрения ERP-системы, в такие активности, как: согласование проектных документов, проведение испытаний системы, обучение сотрудников, мигрирование. Кроме того, они принимают участие в тестовых и продуктивной миграции данных, а также в процедуре бизнес-катовера.

- обновления регламентирующих (регламент закупок, производства, продаж и др., в которых явно упоминается информационная система), кадровых (трудо-вые договора, должностные инструкции, положения и др.) и прочих документов компании (организационная структура, штатная численность и прочие);
- проработки процедуры обработки запросов на изменения, т.е. на дополнительную доработку и развитие КИС после ее успешного продуктивного запуска.

Как видно из описания, данная группа задач не связана явно с каким-либо приложением, а представляет собой ряд инициатив, поэтому уход производителей программного обеспечения практически никак не влияет на нее.

Уровень процессов задается организационной структурой предприятия, а также бизнес-процессами в нем. И первые, и вторые сущности подлежат проектированию в заданной графической нотации как на верхнем, так и нижнем уровне декомпозиции. Нотации широко известны (ARIS VACD, BCM, IDEF0 и ARIS eEPC, BPMN SLD, UML AD, Cross WFD, IDEF3, DFD) и могут быть реализованы в вендорнезависимом ПО, например, в аналогах MS Word, MS PowerPoint и MS Visio. Поэтому уход с российского рынка западных и американских производителей программного обеспечения не влияют существенным образом на этот уровень имплементации.

Следующим уровнем к анализу является уровень данных. Технические особенности построения СУБД (система управления базами данных), а также их установку на сервер баз данных мы рассмотрим на техническом уровне. С точки зрения данных, как было подчеркнuto в работе [5], информация переносится из программной системы отправителя в информационную систему получателя путем нескольких тестовых, а также ранней и последующей продуктивной волн миграций. Следовательно, процесс переноса информации из исторической в целевую систему является больше методологическим нежели вендорзависимым, поэтому он практически не подлежит изменению.

Наконец, уровень технической инфраструктуры, необходимой для функционирования ERP-системы, наиболее сильно страдает от ухода западных производителей. Рассмотрим технический уровень с точки зрения функционирования серверов приложений и баз данных, а также рабочих мест конечных пользователей. В первом случае можно выделить две парадигмы работы серверов: On-Prem и On-Cloud, представимыми отечественными серверами Эльбрус на базе RedHat и СУБД PostgreSQL, а также онлайн решениями от Яндекс, Сбера или VK соответственно. Во втором случае речь

идет о компьютерах пользователей, где также могут быть использованы машины на базе Эльбруса, Байкала и др. В виду крайне ограниченного набора «железа», имеющего по-настоящему Российские корни, технический уровень представляется значительно пострадавшим от действия санкций.

Завершая уровнем приложений, хочется отметить, что ситуация здесь не столь плачевная. Практически все стандарты автоматизации представлены существующими отечественными программными разработками, среди которых можно особенно отметить линейку продуктов от компании 1С, охватывающую практически все направления работы корпоративных функций. Конечно, архитектура, степень кастомизации и разработок в 1С-решениях отличается от западных аналогов, в частности, того же SAP, где продуманы и интегрированы практически все необходимые компоненты совершенно разных по классу информационных систем. Особенность российского программного обеспечения от 1С состоит в том, что оно автоматизирует схожий по охвату набор бизнес-процессов, но делает это большим количеством приложений, которые менее конфигурируемы и более дорабатываемы.

Если объединить полученные результаты, по каждому из уровней внедрения, то получается следующая картина (табл.1): несмотря на кажущуюся зависимость от иностранного ПО, на проект внедрения ERP-систем в большей степени влияет именно «железо», а не софт, все остальные уровни представляют собой методологическую или вендорнезависимую активность, что минимально сказывается на ходе имплементирования. Конечно, в рассмотренном выше примере мы анализировали упрощенный вариант аппаратного обеспечения, но факт остается фактом, именно оно сейчас крайне негативно влияет на ERP-проекты.

Таблица 1. Уровни внедрения ERP-систем и импортозависимость

Уровень внедрения	Использование западных методов и технологий	Степени зависимости от импортных аналогов
Проект	Преимущественно своды знаний PMBoK, Agile Scrum, программное обеспечение MS Project	1 - Низкая (информация по PMBoK, Agile и прочим методологиям общедоступна)
Изменения	Ручная обработка с использованием различных неспециализированных приложений	1 - Низкая (в виду отсутствия специализированного ПО, вопрос больше методологический, чем технический)
Процессы	Графические нотации, представленные как платными программ-	1 - Низкая (по причине наличия офисных приложений,

Уровень внедрения	Использование западных методов и технологий	Степени зависимости от импортных аналогов
	ными пакетами, так и бесплатными и онлайн сервисами (ARIS, UML, IDEF и др.)	способных дублировать функции специализированного ПО)
Данные	Преимущественно методологические вопросы миграции данных	1 - Низкая (вопрос относится больше к методологическому, чем техническому)
Техника	Импортное «железо» от серверов до пользовательских компьютеров, средств коммуникации и общения. Применение аппаратно-программного обеспечения, например, SAP S4 решений	3 - Высокая (российские аналоги имеют урезанный функционал, в ряде случаев вовсе отсутствуют. Потребуется 1-3 лет, чтобы преодолеть столь сильную импортозависимость)
Приложения	Широкий выбор программного обеспечения, позволяющий обрабатывать различные уровни зрелости компании с точки зрения процессов. Решения охватывают широкий набор функций и интегрируемые, как-то продукты от SAP, Microsoft, Oracle, SAS и др.	2 - Средняя (российская линейка программного обеспечения преимущественно представлена 1С-решениями, по многим параметрам отличающаяся от западных образцов. Продукты не хуже аналогов, но другие по природе. Потребуется не менее полу года на изменение привычек пользователей после перехода от западных на российские продукты)

Заключение

Подводя итоги, хочется отметить, что с точки зрения внедрения ERP-систем сложность на данный момент видится не столько в проектировании и внедрении приложений, сколько в аппаратном обеспечении, необходимого для работы приложений. Да, западное ПО и российское отличается, но не стоит забывать, что отечественное программное обеспечение в первую очередь направлено на локальную специфику, как-то регламентированный, а также управленческий учеты, все остальное вторично. За последние несколько десятилетий мы успели привыкнуть к импортным приложениям, но сейчас пришло время начать мыслить по-новому, менять привычки, создавать что-то новое и свое. Важно сделать первый шаг вперед, принимая во внимание,

что предварительно нас ждут несколько шагов назад, однако это единственный способ побороть импортозависимость.

Литература

1. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: теория и практика // Российский технологический журнал. – 2015. – т.8, №3. – с.227-238. – URL: <https://stepanovd.com/science/31-article-2015-2-erpthpr>.
2. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем: учебное пособие. – Ростов н/Д.: Феникс, 2009. – 508 с.
3. Остроух А.В., Суркова Н.Е. Проектирование информационных систем. М.: Лань, 2019. – 164 с.
4. Ширенбек Х., Листер М., Кирмсе Ш. Руководство к своду знаний по управлению проектами. Руководство РМВоК. Шестое издание. – М.: Олимп-Бизнес, 2019. – 974 с.
5. Еременко Я.О. Особенности миграции данных в SAP ERP // Корпоративные информационные системы. – 2019. – №3(7). – С. 22-28. – URL: <https://corpinfosys.ru/archive/issue-7/67-2019-7-migration>.

Выходные данные статьи

Степанов Д.Ю. Импортзамещение в проектах внедрения ERP-систем // Корпоративные информационные системы. – 2022. – №2 (18) – С. 36-41. – URL: <https://corpinfosys.ru/archive/issue-18/200-2022-18-erpimportsubstitution>.

Об авторе



Степанов Дмитрий Юрьевич – кандидат технических наук, доцент МИРЭА, принимал участие более чем в 10 проектах внедрения корпоративных информационных систем на базе SAP, Microsoft и Sage. Специализируется на управлении материальными потоками, сбытом и системой документов. Автор более 25 статей, в том числе в «Логистика сегодня», «Проблемы экономики», «САТтер». Электронный адрес: mail@stepanovd.com.