



КИС

# КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Солдатов С.А.

Обзор открытой корпоративной информационной системы Odoo

Степанов Д.Ю., Вельсовский А.В.

Применение Agile Scrum в проектах SAP

Гудков Е.А., Деревнина А.М., Катасонова Н.С.

Анализ каскадной, итерационной и спиралевидной моделей  
внедрения корпоративных информационных систем

Степанов Д.Ю.

Особенности внедрения проекта MRP по точке перезаказа

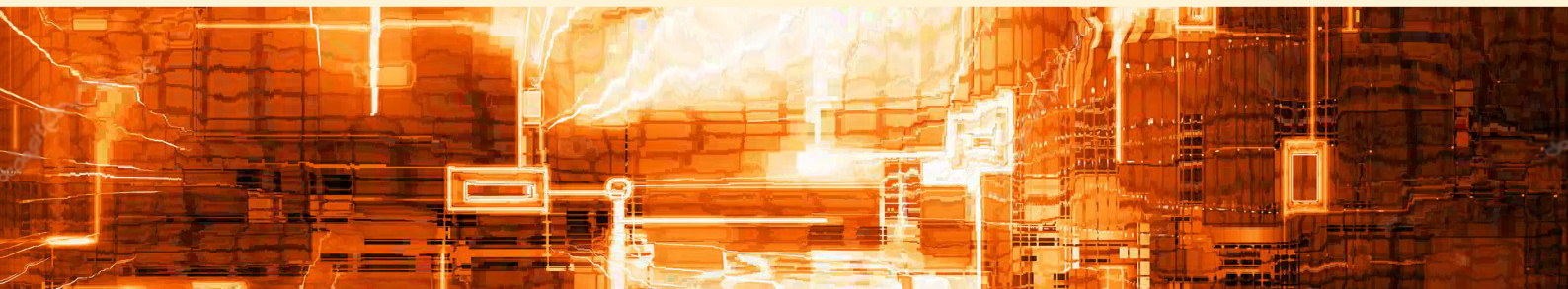
Терентьев И.М.

Сложности внедрения WMS-систем на примере SAP ERP

[corpinfosys.ru](http://corpinfosys.ru)

Научно-популярный сетевой журнал

Выпуск №1  
2018 год



## 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:

- 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 членам редакционной комиссии и авторам первого выпуска журнала.

Прошло около двух лет от идеи до создания сетевого ресурса corpinfosys.ru. Первый выпуск издания является пилотным в терминах ERP-систем и содержит подборку статей о бесплатной корпоративной информационной системе Odoo, сложностях внедрения систем управления складами и функционала планирования потребности в материалах, а также моделях внедрения информационных систем.

Со временем издание претерпит значительные изменения для обеспечения максимального удобства, доступности и открытости всем желающим. Выражаю благодарность тем, кто способствовал и помогал в подготовке выхода журнала, в особенности Кристине Трухан, не увидевшей выпуск 1-го номера.

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

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



### Обзор современных корпоративных информационных систем

*Солдатов С.А.*

Обзор открытой корпоративной информационной системы Odoo.....1

### Управление проектом внедрения корпоративных информационных систем

*Степанов Д.Ю., Вельсовский А.В.*

Применение Agile Scrum в проектах SAP.....9

*Гудков Е.А., Деревнина А.М., Катасонова Н.С.*

Анализ каскадной, итерационной и спиралевидной моделей  
внедрения корпоративных информационных систем.....18

### Опыт внедрения корпоративных информационных систем

*Степанов Д.Ю.*

Особенности проекта внедрения MRP по точке перезаказа.....30

*Терентьев И.М.*

Сложности внедрения WMS-систем на примере SAP ERP.....40

# Обзор открытой корпоративной информационной системы Odoo

Солдатов Сергей Александрович

**Аннотация:** в статье описывается свободно распространяемая ERP-система Odoo, завоевавшая популярность своей открытостью и широкой поддержкой среди независимых разработчиков. Дается общее представление о системе, её структуре и возможных сферах применения.

Свободно распространяемое ПО уже много лет активно используется частными лицами, государственными и образовательными учреждениями, крупными корпорациями и малым бизнесом. Преимущества очевидны: низкая цена установки (а зачастую и вовсе бесплатно), сравнительно широкая поддержка популярных продуктов сообщества разработчиков, наличие исходных кодов и как следствие возможность доработать ПО под свои требования, не дожидаясь обратной связи от разработчиков.

Появление на рынке ERP и CRM-систем свободно распространяемых продуктов было делом времени. Одним из таких продуктов стала система Odoo (ранее именуемая Tiny ERP и OpenERP).

## История системы

Как и большинство крупных Open-проектов, история Odoo началась со студентов Фабиана Пинкарса (Fabien Pinckaers) и Антони Лесусса (Antony Lesuisse) [1]. В 2000 году ими была написана простая система управления предприятием на языке Python, предназначенная для внедрения на небольших бельгийских фирмах, работающих в основном в сфере услуг и дистрибуции [2].

Изначально система называлась Tiny ERP. Фабиан Пинкарс совмещал разработку и внедрение системы, но уже с 2004 в компании Tiny появились наёмные сотрудники. В 2006 году был запущен ресурс TinyForge, а к разработке подключены сторонние независимые программисты. С 2009 года система становится частью дистрибутивов Ubuntu и Debian, в мае 2014 года она переименована в Odoo, а компания - в Odoo S.A.

На текущий момент в фирме Odoo S.A. работает более 250 сотрудников, она имеет 6 офисов (в Бельгии, Люксембурге, Индии, Гонконге, два офиса в США - в Сан-Франциско и Нью-Йорке), а в сообществе независимых разработчиков зарегистрировано более 1500 участников. Программное обеспечение загружается 1000 раз в день, и существует более 14000 программных модулей. По заявлению разработчиков Odoo, это самое устанавливаемое программное обеспечение для

бизнеса в мире. Сегодня насчитывается около 2 000 000 пользователей Odoo по всему миру, а более 550 компаний-партнёров в 120 странах занимаются распространением и внедрением системы.

Odoo успешно используется в таких известных в России международных компаниях как Auchan, Danone, Hyundai и подразделениях Toyota. В октябре 2017 года вышла одиннадцатая версия системы.

### Технические требования и структура системы

Система не требовательна к аппаратному и программному обеспечению [3]. И это понятно, у небольшой компании или частного предпринимателя (да-да, даже частный предприниматель может найти ей применение) нет больших возможностей для создания солидной ИТ-инфраструктуры, которую требуют SAP или Microsoft Dynamics AX. В табл. 1 приведены системные требования. Помимо указанной в таблице ОС Ubuntu, Odoo прекрасно работает и под Windows, а также с другими Linux-дистрибутивами.

**Таблица 1.** Системные требования Odoo к серверной платформе

| Общие требования к серверу                                                                                                                                                                                 | Минимальные требования к серверу                                                                             | Рекомендуемая конфигурация сервера                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Для 5 пользователей<br>= 2×ЦПУ, 2 Гбайт ОЗУ<br>Для 20 пользователей<br>= 4×ЦПУ, 8 Гбайт ОЗУ<br>Для 100 и более<br>пользователей<br>= отдельные сервер<br>приложения и БД, каждый<br>с 8×ЦПУ с 32 Гбайт ОЗУ | ЦПУ = Intel x64<br>Количество ЦПУ = 1<br>Количество ядер в ЦПУ = 6<br>ОЗУ = 8 Гбайт<br>ОС = Ubuntu 14.04 LTS | ЦПУ = Intel x64<br>Количество ЦПУ = 2<br>Количество ядер в ЦПУ = 8<br>ОЗУ = 8 Гбайт<br>ОС = Ubuntu 14.04<br>HDD = размер зависит от<br>количества клиентов |

Так как система написана на базе языке Python, то для работы системы в обязательном порядке надо установить комплект библиотек данного языка. Структура системы максимально упрощена и включает три компонента (рис. 1): СУБД PostgreSQL, сервер приложений и web-сервер. Первый служит для хранения всех данных системы и части настроек, второй реализует бизнес-логику конкретного предприятия, а третий предоставляет доступ через web-браузер ко второму компоненту. В зависимости от объёма решаемых задач все части системы могут быть установлены на одном сервере или размещены на разных.

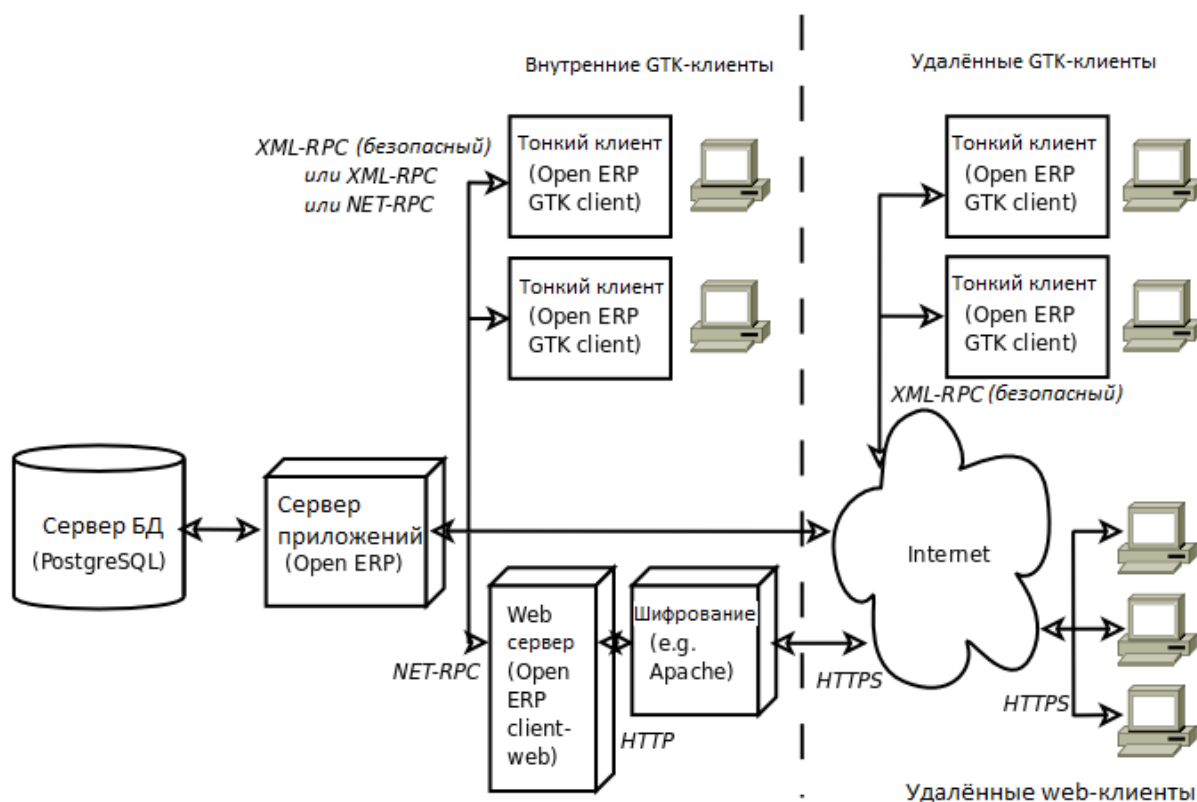


Рис. 1. Структура системы Odoo

Доступ к серверу приложений возможен не только из web-браузера, но и из графического GTK-клиента Odoo. Тем не менее разработчики Odoo использовать web-интерфейс, хотя и признают, что из-за кэширования в web-браузере возможны проблемы при изменении элементов системы: изменения сделаны, но пользователи не сразу видят их. Будучи в русле современных тенденций, Odoo сейчас предлагает и облачный вариант, когда, по сути, не требуется ничего ставить локально, а предлагается воспользоваться уже готовой инфраструктурой.

### Сферы применения

Сферы применения Odoo сравнительно обширны: CRM, продажи и покупки, склад, производство, управление персоналом, бухгалтерский учёт, документооборот и ведение корпоративного портала. Следует отметить, что Odoo имеет широкие средства для ведения бизнеса в интернете. Так, среди приложений есть редактор сайтов (рис. 2), приложения для онлайн-магазинов, и средства для организации онлайн-поддержки клиентов. Есть специальные модули для управления проектами, контроля качества, а также проведения маркетинговых мероприятий.

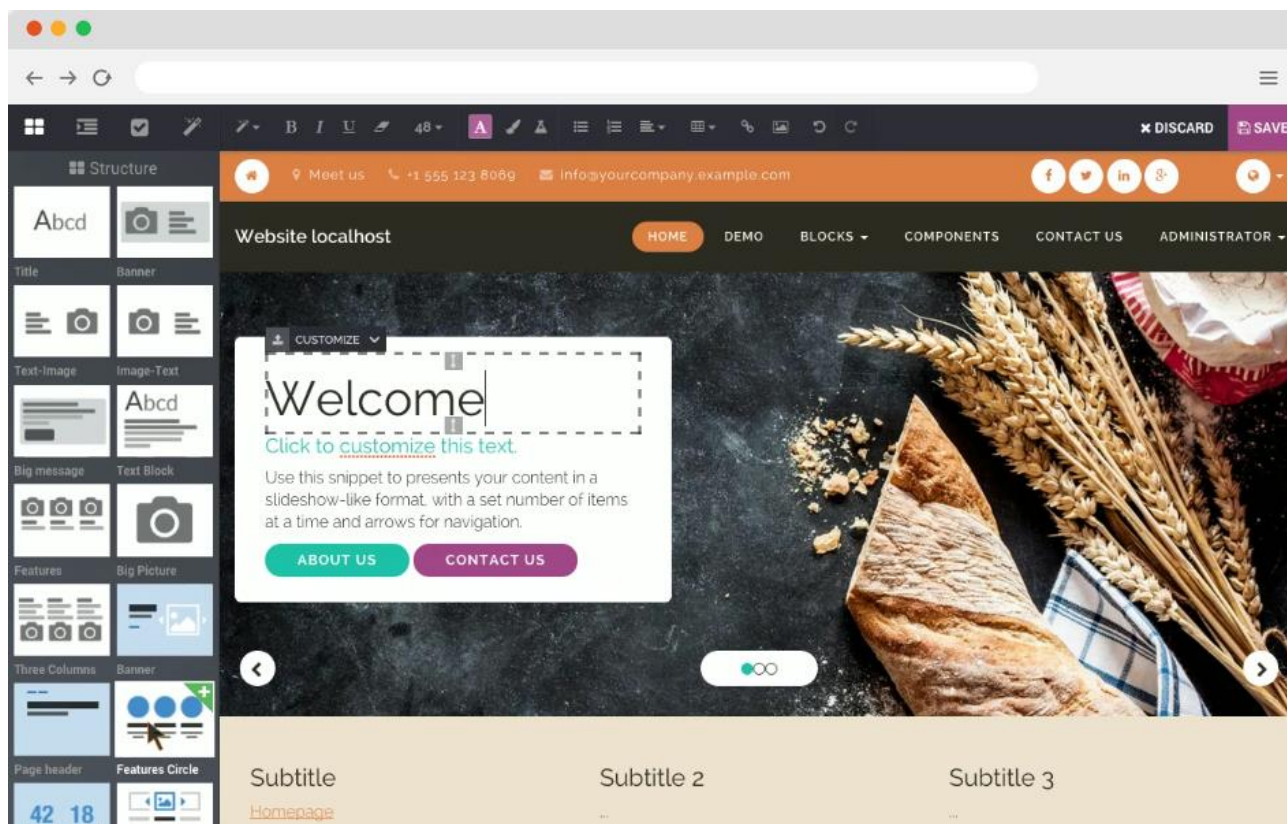


Рис. 2. Web-редактор Website builder

### Состав модулей

Состав модулей (в терминах Odoo - Apps) обширен благодаря широкому сообществу сторонних разработчиков. Модули есть как бесплатные [4], так и платные, обслуживаемые в основном по подписке. Базовый пакет из 28 модулей обойдётся в 120 евро в год [5] на одного пользователя (рис. 3), но он будет включать все необходимые модули для ведения бизнеса небольшой компании:

- CRM – управление клиентами;
- Website – создание и управление сайтом компании;
- Project – проектное управление;
- Equipment – инвентаризация материально-технических средств компании;
- Inventory – управление складом;
- Sales – управление продажами;
- PLM – управление жизненным циклом изделий;
- Accounting – бухгалтерский учёт;
- Helpdesk – техническая поддержка заказчиков.

Возможно бесплатное тестирование в течение 15 дней, а также помесечная подписка (цена будет немного выше).

## Odoo Pricing

### Choose the number of Users

1 Users x 10,00 EUR/Month **12,00 EUR/Month**

| Annually          | Monthly  |
|-------------------|----------|
| 1 Users           | 0,00 EUR |
| User Discount (*) | 0,00 EUR |
| 1 Apps            | 0,00 EUR |
| Total / year      | 0,00 EUR |

### Choose your Apps

|                                                                                  |                                                                          |                                                                            |
|----------------------------------------------------------------------------------|--------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <b>CRM</b><br>12,00 EUR / month <input checked="" type="checkbox"/>              | <b>Invoicing</b><br>6,00 EUR / month <input type="checkbox"/>            | <b>Sales</b><br>6,00 EUR / month <input type="checkbox"/>                  |
| <b>Website</b><br>12,00 EUR / month <input type="checkbox"/>                     | <b>eCommerce</b><br>6,00 EUR / month <input type="checkbox"/>            | <b>Point of Sale</b><br>12,00 EUR / month <input type="checkbox"/>         |
| <b>Accounting</b><br>12,00 EUR / month <input type="checkbox"/>                  | <b>Project</b><br>12,00 EUR / month <input type="checkbox"/>             | <b>Inventory</b><br>18,00 EUR / month <input type="checkbox"/>             |
| <b>Manufacturing</b><br>24,00 EUR / month <input type="checkbox"/>               | <b>Purchase</b><br>6,00 EUR / month <input type="checkbox"/>             | <b>Timesheet</b><br>6,00 EUR / month <input type="checkbox"/>              |
| <b>Email Marketing</b><br>6,00 EUR / month <input type="checkbox"/>              | <b>Expenses</b><br>6,00 EUR / month <input type="checkbox"/>             | <b>Events</b><br>6,00 EUR / month <input type="checkbox"/>                 |
| <b>Leaves</b><br>6,00 EUR / month <input type="checkbox"/>                       | <b>Recruitment</b><br>6,00 EUR / month <input type="checkbox"/>          | <b>Appraisal</b><br>6,00 EUR / month <input type="checkbox"/>              |
| <b>Subscription</b><br>12,00 EUR / month <input type="checkbox"/>                | <b>Sign</b><br>12,00 EUR / month <input type="checkbox"/>                | <b>Equipment</b><br>6,00 EUR / month <input type="checkbox"/>              |
| <b>Quality Control</b><br>12,00 EUR / month <input type="checkbox"/>             | <b>Studio</b><br>36,00 EUR / month <input type="checkbox"/>              | <b>Helpdesk</b><br>12,00 EUR / month <input type="checkbox"/>              |
| <b>Maintenance (Manufacturing)</b><br>12,00 EUR / month <input type="checkbox"/> | <b>PLM (Manufacturing)</b><br>12,00 EUR / month <input type="checkbox"/> | <b>Appointment Scheduling</b><br>6,00 EUR / month <input type="checkbox"/> |
| <b>Marketing Automation</b><br>18,00 EUR / month <input type="checkbox"/>        |                                                                          |                                                                            |

Those apps are free as long as you don't need more apps or hosting options.

[start now](#)

(\*) New customers get a discount on the initial number of users purchased. (10,00 EUR instead of 12,00 EUR).

### Extra Integrations

|                                                          |                                                          |                                                           |
|----------------------------------------------------------|----------------------------------------------------------|-----------------------------------------------------------|
| <b>DHL</b><br>18,00 EUR / month <input type="checkbox"/> | <b>UPS</b><br>18,00 EUR / month <input type="checkbox"/> | <b>USPS</b><br>18,00 EUR / month <input type="checkbox"/> |
|----------------------------------------------------------|----------------------------------------------------------|-----------------------------------------------------------|

Рис. 3. Выбор модулей Odoo для подписки

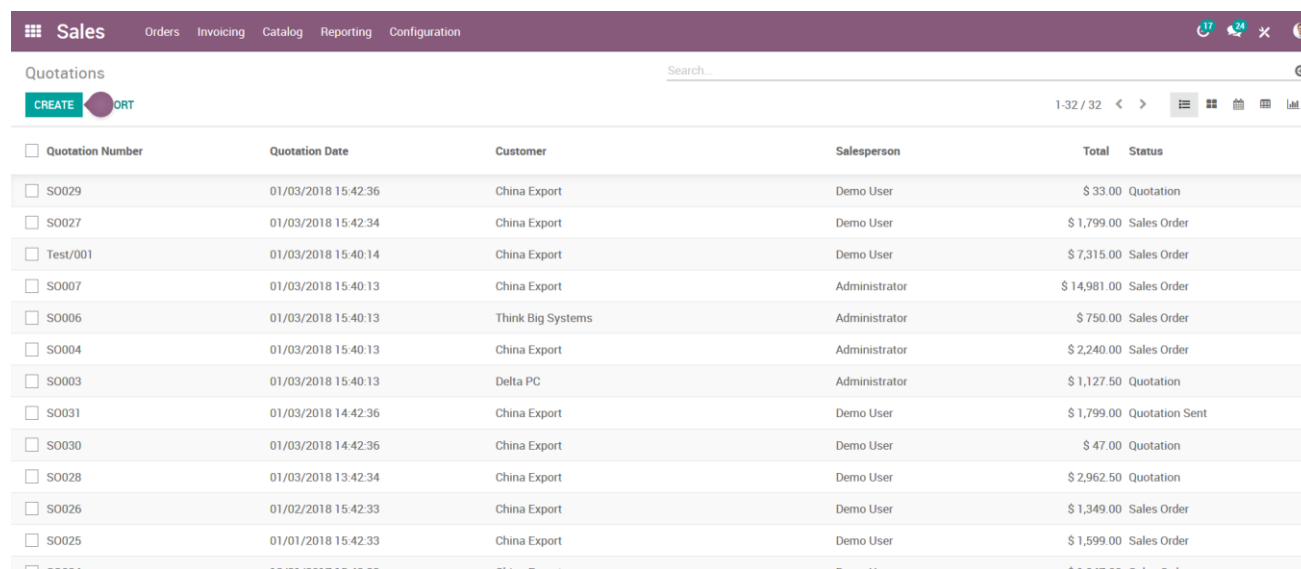
Есть и небольшая «ложка дёгтя». Некоторые модули взаимосвязаны и нельзя купить один, не купив другой. Например, нельзя купить модуль для управления продажами «Sales» не купив «Invoicing» для выставления счетов. Логически всё обосновано, но главное помнить об этом при планировании бюджета. Важно отметить, что не все пользователи учитываются в подписке, а только те, кто имеет права на создание и редактирование данных в системе. Так, ваши клиенты и сотрудники, которые могут только смотреть данные, в том числе и посетители созданного в Odoo сайта, не будут учитываться в подписке.

Возможно и вовсе ничего не покупать, общедоступная версия Odoo содержит меньше модулей, но все они бесплатны. Единственный минус - в бесплатном

комплекте нет среды разработки новых модулей, но их разработка возможна, так как для всех бесплатных модулей доступны исходные коды. Также не предусмотрена техническая поддержка в режиме 24/7.

### Знакомство с системой

Для знакомства с системой на сайте Odoo.com есть раздел Demo [6], где можно узнать возможности стандартного набора модулей. К сожалению, в демо-режиме нет русского языка, как, собственно, и любого другого, кроме английского. На стартовом экране размещены иконки модулей, каждый из них можно запустить. Демонстрационная база данных хоть и скудна, но позволяет получить общее представление о системе (рис. 4).



The screenshot shows the Odoo Sales module interface. At the top, there's a navigation bar with 'Sales' selected, and other options like 'Orders', 'Invoicing', 'Catalog', 'Reporting', and 'Configuration'. Below the navigation bar, there's a search bar and a 'CREATE' button. The main area displays a table of quotations with columns: Quotation Number, Quotation Date, Customer, Salesperson, Total, and Status. The table contains 14 rows of data, including quotations for 'China Export', 'Think Big Systems', and 'Delta PC'.

| Quotation Number | Quotation Date      | Customer          | Salesperson   | Total        | Status         |
|------------------|---------------------|-------------------|---------------|--------------|----------------|
| SO029            | 01/03/2018 15:42:36 | China Export      | Demo User     | \$ 33.00     | Quotation      |
| SO027            | 01/03/2018 15:42:34 | China Export      | Demo User     | \$ 1,799.00  | Sales Order    |
| Test/001         | 01/03/2018 15:40:14 | China Export      | Demo User     | \$ 7,315.00  | Sales Order    |
| SO007            | 01/03/2018 15:40:13 | China Export      | Administrator | \$ 14,981.00 | Sales Order    |
| SO006            | 01/03/2018 15:40:13 | Think Big Systems | Administrator | \$ 750.00    | Sales Order    |
| SO004            | 01/03/2018 15:40:13 | China Export      | Administrator | \$ 2,240.00  | Sales Order    |
| SO003            | 01/03/2018 15:40:13 | Delta PC          | Administrator | \$ 1,127.50  | Quotation      |
| SO031            | 01/03/2018 14:42:36 | China Export      | Demo User     | \$ 1,799.00  | Quotation Sent |
| SO030            | 01/03/2018 14:42:36 | China Export      | Demo User     | \$ 47.00     | Quotation      |
| SO028            | 01/03/2018 13:42:34 | China Export      | Demo User     | \$ 2,962.50  | Quotation      |
| SO026            | 01/02/2018 15:42:33 | China Export      | Demo User     | \$ 1,349.00  | Sales Order    |
| SO025            | 01/01/2018 15:42:33 | China Export      | Demo User     | \$ 1,599.00  | Sales Order    |
| SO024            | 12/31/2017 15:42:33 | China Export      | Demo User     | \$ 1,087.00  | Sales Order    |

Рис. 4. Модуль Sales с демо-данными

Нажав на пиктограмму Apps, можно получить доступ к панели с более подробной информацией о каждом приложении, а также удалить или установить различные приложения. Также на демо-сайте возможно «поиграть» с интерфейсом модулей, что-то удалить, что-то добавить или перекомпоновать (рис. 5). Для этого надо нажать на кнопку с рисунком в виде гаечного ключа и карандаша в правом верхнем углу.

На случай, если есть желание поработать уже с реальными данными, на сайте Odoo.com есть раздел Trial [7], где предлагается выбрать одно приложение с неограниченным количеством пользователей совершенно бесплатно, но уже самому наполнить его базу данных.

| Quotation Number | Quotation Date      | Customer          | Salesperson   | Total        | Status         |
|------------------|---------------------|-------------------|---------------|--------------|----------------|
| S0029            | 01/03/2018 15:42:36 | China Export      | Demo User     | \$ 33.00     | Quotation      |
| S0027            | 01/03/2018 15:42:34 | China Export      | Demo User     | \$ 1,799.00  | Sales Order    |
| Test/001         | 01/03/2018 15:40:14 | China Export      | Demo User     | \$ 7,315.00  | Sales Order    |
| S0007            | 01/03/2018 15:40:13 | China Export      | Administrator | \$ 14,981.00 | Sales Order    |
| S0006            | 01/03/2018 15:40:13 | Think Big Systems | Administrator | \$ 750.00    | Sales Order    |
| S0004            | 01/03/2018 15:40:13 | China Export      | Administrator | \$ 2,240.00  | Sales Order    |
| S0003            | 01/03/2018 15:40:13 | Delta PC          | Administrator | \$ 1,127.50  | Quotation      |
| S0031            | 01/03/2018 14:42:36 | China Export      | Demo User     | \$ 1,799.00  | Quotation Sent |
| S0030            | 01/03/2018 14:42:36 | China Export      | Demo User     | \$ 47.00     | Quotation      |
| S0028            | 01/03/2018 13:42:34 | China Export      | Demo User     | \$ 2,962.50  | Quotation      |
| S0026            | 01/02/2018 15:42:33 | China Export      | Demo User     | \$ 1,349.00  | Sales Order    |
| S0025            | 01/01/2018 15:42:33 | China Export      | Demo User     | \$ 1,599.00  | Sales Order    |
| S0024            | 12/31/2017 15:42:33 | China Export      | Demo User     | \$ 1,047.00  | Sales Order    |
| S0023            | 12/30/2017 15:42:33 | China Export      | Demo User     | \$ 1,199.00  | Sales Order    |
| S0022            | 12/29/2017 15:42:33 | China Export      | Demo User     | \$ 750.00    | Sales Order    |
| S0021            | 12/28/2017 15:42:32 | China Export      | Demo User     | \$ 900.00    | Sales Order    |
| S0020            | 12/27/2017 15:42:32 | China Export      | Demo User     | \$ 599.00    | Sales Order    |

Рис. 5. Модуль Sales в режиме редактирования интерфейса

## Odoo в России

В России Odoo известна довольно недавно, основные проекты, о которых можно найти информацию, запущены в 2012 году и позже. Тем не менее, среди тех, кто установил Odoo самостоятельно или со сторонней помощью есть как производственные, так и логистические фирмы, торговые компании, рекламные агентства и даже монтажные организации и автосервисы [8, 9].

Для обмена опытом российскими разработчиками под Odoo запущен сайт <http://odoo-russia.ru/>. И хотя по сравнению с зарубежными ресурсами про Odoo активность его ещё незначительна, тем не менее на нём зарегистрировано более 5000 пользователей. Не забыл о России и головной офис Odoo: существует сайт <https://odoo.ru> с материалами, а также демо-примерами проектов на русском языке (<https://demo.odoo.ru/>).

## Литература

1. The Tryton Project [Электронный ресурс] // Сайт конференции IWEEE - Режим доступа: [http://www.iweee.org/2015-las\\_palmas/presentations/files/Nicoe-1-IWEEE2015.pdf](http://www.iweee.org/2015-las_palmas/presentations/files/Nicoe-1-IWEEE2015.pdf) (дата обращения 17.12.2017)
2. Odoo [Электронный ресурс] // Сайт Wikipedia - Режим доступа: <https://ru.wikipedia.org/wiki/Odoo> (дата обращения 17.12.2017)
3. Server requirements [Электронный ресурс] // Сайт компании Odoo S.A. - Режим доступа: <https://www.odoo.com/forum/help-1/question/server-requirements-121217> (дата обращения 17.12.2017)

4. Odoo Free Apps [Электронный ресурс] // Сайт компании Odoo S.A. - Режим доступа: <https://www.odoo.com/apps?price=Free> (дата обращения 18.12.2017)
5. Odoo Pricing [Электронный ресурс] // Сайт компании Odoo S.A. - Режим доступа: [https://www.odoo.com/pricing#pl=77&num\\_users=1&hosting=online&implementation\\_service=self&pack=25&force\\_country=RU&integrating\\_partner\\_id=0&price\\_by=yearly](https://www.odoo.com/pricing#pl=77&num_users=1&hosting=online&implementation_service=self&pack=25&force_country=RU&integrating_partner_id=0&price_by=yearly) (дата обращения 19.12.2017)
6. Odoo Demo [Электронный ресурс] // Сайт компании Odoo S.A. - Режим доступа: <https://demo3.odoo.com/web#home> (дата обращения 04.01.2018)
7. Odoo Online Trial [Электронный ресурс] // Сайт компании Odoo S.A. - Режим доступа: <https://www.odoo.com/trial> (дата обращения 05.01.2018)
8. Узкопрофильные отраслевые решения на основе Odoo [Электронный ресурс] // Сайт российского сообщества разработчиков и пользователей Odoo - Режим доступа: <http://odoo-russia.ru/forum/vertical/> (дата обращения 10.01.2018)
9. Примеры внедрения Odoo ERP [Электронный ресурс] // Сайт компании IT Libertas - Режим доступа: [https://itlibertas.com/ru\\_RU/page/odoo-portfolio](https://itlibertas.com/ru_RU/page/odoo-portfolio) (дата обращения 10.01.2018)

### **Выходные данные статьи**

Солдатов С.А. Обзор открытой корпоративной информационной системы Odoo  
// Корпоративные информационные системы. - 2018. - №1. - С. 1-8. - URL:  
<http://corpinfosys.ru/archive/issue-1/44-2018-1-odoo>

### **Об авторе**



**Солдатов Сергей Александрович** - кандидат технических наук, занимается разработкой и внедрением комплексных информационных систем класса SCADA и MES. Является автором более 20 научных статей, в том числе публикаций в журналах «Современные технологии автоматизации», «Информационные системы и технологии», «В мире научных открытий» и др. Электронная почта: [ssacompany@mail.ru](mailto:ssacompany@mail.ru)

## Применение Agile Scrum в проектах SAP

Степанов Дмитрий Юрьевич,  
Вельсовский Андрей Вельтерович

**Аннотация:** работа содержит описание метода Agile Scrum в проектах внедрения корпоративных информационных систем на базе SAP. Рассматриваются вопросы применения Scrum в проектах развития, тиражирования и внедрения «с нуля» решений SAP, а также использования гибкой методологии Agile для разработки и кастомизации информационных систем. Сделан вывод о целесообразности применения метода Scrum в проектах развития SAP-систем путём доработки.

Пожалуй, нет более популярной темы для обсуждения, чем применение Agile в проектах SAP. Несмотря на то, что принципы гибкой разработки были сформулированы ещё в 2001 году [1], их использование в настоящее время становится как никогда востребованным. Связано это в первую очередь с тем, что последнее десятилетие знаменуется массовым использованием информационных технологий (далее – ИТ) в повседневной жизни: порталы государственные услуг, интернет-магазины, электронное правительство и многое другое. Вышесказанное требует как грамотной разработки программного обеспечения (далее – ПО), так и не менее искусного его внедрения.

|          |                                                   |                                     |                                                                      |                                                                 |
|----------|---------------------------------------------------|-------------------------------------|----------------------------------------------------------------------|-----------------------------------------------------------------|
| Ценности | Люди и взаимодействие между ними важнее процессов | Рабочий продукт важнее документации | Сотрудничество с заказчиком важнее обсуждения условий договора с ним | Готовность к изменениям важнее следования первоначальному плану |
|          | Ранняя поставка продукта                          | Изменения приветствуются            | Регулярные поставки                                                  | Бизнес и команда работают вместе                                |
|          | Команде доверяют и поддерживают                   | Общение лицом к лицу                | Работающий продукт                                                   | Поддержание постоянного темпа                                   |
|          | Внимание на качество                              | Минимизация лишней работы           | Самоорганизация команды                                              | Непрерывное совершенствование                                   |

Рис. 1. Ценности и принципы Agile

Agile представляет собой методологию реализации и внедрения ПО на основе итерационной модели и включает совокупность методов, к которым можно отнести FDD (Feature Driven Development – разработка, управляемая функциональностью), XP (eXtreme Programming – экстремальное программирование), Kanban, Crystal и др. Суть методологии заключается в использовании 4 базовых ценностей и 12 принципов (рис. 1), объявленных в манифесте Agile [2], следование которым призвано существенно облегчить имплементацию информационных систем (далее – ИС). Одним из ярких примеров использования принципов Agile является метод Scrum. Рассматриваемый как противовес классической каскадной модели (Waterfall – водопадная модель) внедрения ИС, метод Scrum даёт чёткое представление процесса имплементации и описывает реализацию базовых составляющих манифеста. Справедливости ради следует отметить, что Scrum частично применяется и в водопадной модели, например, для уточнения требований на фазе анализа путём прототипирования.

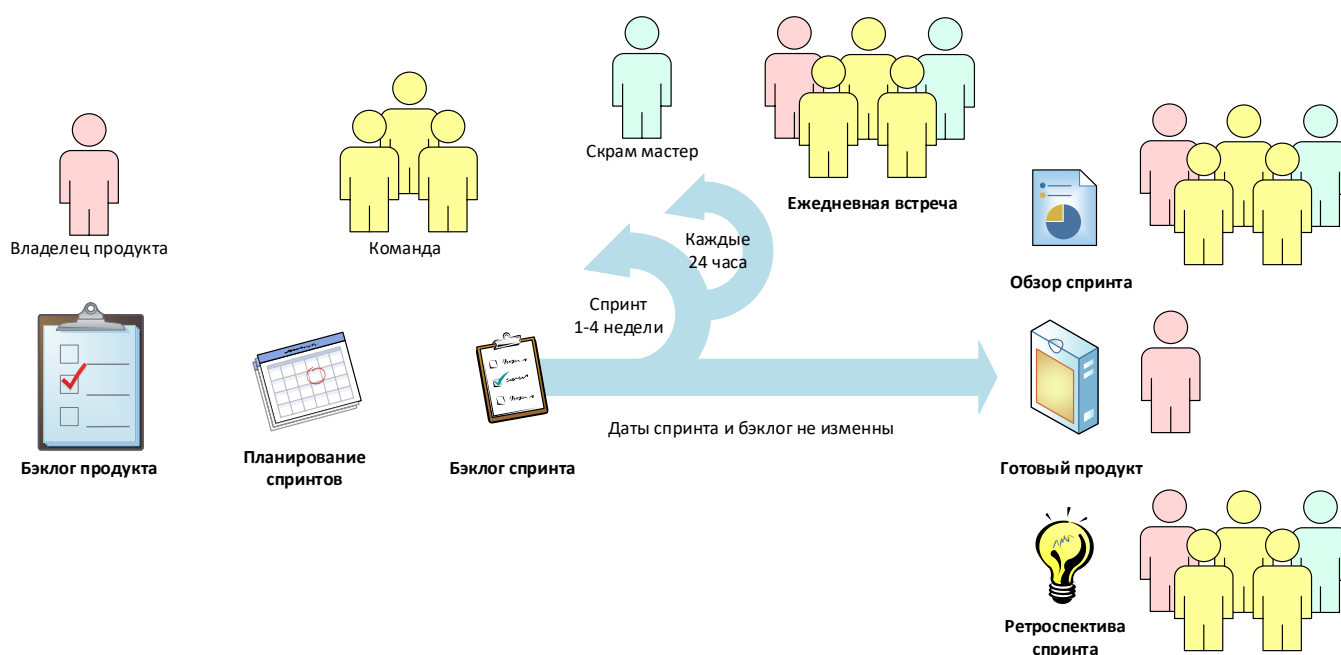


Рис. 2. Процесс реализации проекта согласно Scrum

Напомним, итеративный подход реализации ПО заключается в разбиении процесса внедрения на стадии, называемые итерациями, в рамках которых разрабатывается и демонстрируется заказчику реализованная часть решения [3]. При этом как таковые требования вообще могут отсутствовать, количество предстоящих итераций не известно, а объём проекта изменяем при фиксированных сроках и

бюджете. Следуя методу Scrum, итерации называют спринтами, список требований – бэклогом (Backlog), ход проекта контролируют на доске стикерами, а команда рассматривается как самоорганизующаяся [4]. Концептуальная картина ведения проекта имплементации согласно Scrum дана выше на рис. 2.

Несмотря на отличие каскадной и итерационной моделей внедрения ИС, в обоих подходах подтверждением работоспособности разработанной системы служит успешно пройденное приёмочное тестирование (User Acceptance Test – UAT) [3]. Однако метод Scrum существенно отличается от обеих моделей из-за отсутствия UAT и документирования решения. Проект внедрения корпоративных информационных систем (далее – КИС) с использованием Scrum состоит из следующих шагов:

- идентификация и анализ требований, предъявляемых к КИС, приоритезация найденных требований и формирование бэклога продукта;
- определение числа и продолжительности спринтов разработки КИС; формирование бэклога спринтов и их распределение по итерациям;
- реализация КИС согласно бэклогу спринта, функциональное и интеграционное тестирование, демонстрация полученного результата владельцу продукта и заказчику, ретроспектива спринта и обновление бэклогов, а также продуктивная эксплуатация реализованного решения (для всех спринтов) [4].

Более того, в отличие от каскадной и итерационных моделей, мерилем реализации проекта Scrum является спринт (часть продукта), а не продукт. Так в Scrum промышленному использованию подлежит результат каждого спринта, даже если финальный спринт, обозначающий реализацию всего продукта, ещё не реализован.

Методология Agile изначально была ориентирована на разработку, но не на кастомизацию ПО, именно поэтому под Scrum-командой понимается состав из 5-7 разработчиков. Кастомизация представляет собой настройку ИС, не требующую программной доработки решения. Настройка ПО ведётся силами функциональных консультантов, в то время как реализация решения – программными разработчиками. Следует отметить, что число вариаций настроек системы под нужды заказчика весьма ограничено. Кастомизация ИС обеспечивает стандартизацию, унификацию, а также прозрачность решения: новичку гораздо проще разобраться в настройках в виду их детерминированности, нежели с программными разработками. Допустим, кастомизация ведётся силами разработчиков. Современные корпоративные информационные системы, например, от компании SAP AG: ERP, SRM, SRM и другие, функционируют и претерпевают изменения из-за:

- внедрения «с нуля»;
- тиражирования;
- развития,

а проект их внедрения характеризуется функциональными:

- большим объёмом основных и переменных данных для целей миграции и последующего использования в SAP;
- единой интегрированной организационной структурой предприятия в SAP;
- заданным числом бизнес-процессов для отражения в SAP;
- большим массивом транспортных запросов и задач для переноса в SAP

и организационными особенностями:

- большим числом членов проектной команды (в зависимости от проекта - 1-5 человек в рамках одного функционального направления, в проекте может быть от 1 до 7 направлений);
- интеграцией SAP с внешними системами.

Внедрение подобных систем осуществляется преимущественно на основе каскадной модели в виду дороговизны, объёмности, сложности и продолжительности проекта. Прделаем следующее упражнение: выделим способы реализации КИС, а также виды проектов, далее для пары «способ-вид» попытаемся эмпирически оценить целесообразность использования Agile Scrum (табл. 1). В качестве оценивания выберем шкалу от 1 до 3, где 1 - применение маловероятно, 2 - использование возможно, но требуется значительная трансформация команды, 3 - рекомендуется использовать Scrum.

**Таблица 1.** Целесообразность применения Agile Scrum в проектах SAP

| Способ реализации      | Вид проекта                       | Целесообразность использования Agile Scrum |
|------------------------|-----------------------------------|--------------------------------------------|
| Разработка             | Развитие                          | 2-3                                        |
| Настройка              | Развитие                          | 1-2                                        |
| Настройка и разработка | Внедрение «с нуля», тиражирование | 1                                          |

Использование Agile Scrum для доработки SAP системы в процессе её развития действительно может позволить достичь положительного эффекта. Стоит отметить, что разработка затрагивает лишь ограниченный функционал SAP-системы, сравнимый по объёму с модулем, например, закупки, сбыт, запасы и др.

Применение Scrum диктует особые требования к организации и составу команды: теперь архитектурные решения должны приниматься доверительно и коллективно; разработка ведётся от требований, проектные решения, спецификации на разработку и прочая документация отсутствуют; каждый ежедневно отчитывается о результатах проделанной работы. Для использования гибкой методологии необходима трансформация проектной команды, первая попытка которой может закончиться неудачей. Преимущества Scrum будут ощутимы только при условии, что команда непрерывно работает над проектами, используя гибкую методологию Agile, а не от случая к случаю: применение Scrum требует преодоления инерции водопадной модели. Одно и то же требование может быть одновременно неправильно трактовано и неверно реализовано, непрерывная демонстрация разрабатываемого решения владельцу продукта и конечным пользователям позволяет получить именно тот результат, который ожидает и хочет увидеть пользователь. В этом заключается преимущество итерационной модели, к которой относится и Agile. Таким образом, целесообразность использования Agile Scrum в проектах развития SAP-систем за счёт их доработки видится оправданной, однако требует значительных изменений в мышлении, организации и понимании новых правил игры всей проектной команды (табл. 1, оценка целесообразности 2-3).

Обратимся к случаю кастомизации системы SAP в процессе её развития. Самый простой пример – внедрение ранее не использованного модуля, например, SAP ERP WM (Warehouse Management – управление складами). Подобные активности не требуют чрезмерных трудозатрат функционального консультанта. Давайте разберемся за счёт чего достигается положительный эффект применения Scrum в процессе доработки SAP. Программа может быть организована и запрограммирована огромным числом способов, гибкая модель разработки позволяет выбрать лишь тот, который удовлетворяет конечному пользователю. В случае настройки SAP-системы ситуация в корне иная: кастомизация решения весьма ограничена, а варианты настройки либо носят единичный характер, либо очень близки. Демонстрация промежуточного продукта пользователю кардинально не может изменить выбранного решения в виду его детерминированности. В итоге, конечный пользователь мало на что может повлиять, а сам метод Scrum обеспечивает лишь быструю доставку решения без какой-либо существенной обратной связи (табл. 1., оценка целесообразности 1-2).

И, наконец, полномасштабное внедрение SAP. Значительная часть проектов как тиражирования, так имплементации SAP «с нуля» требует доработки: кастомизация в большинстве своём не может покрыть всех требований бизнеса. Начнём с

организационных сложностей: проект внедрения SAP требует вовлечения большого числа как разработчиков, так и консультантов. В зависимости от проекта количество членов команды может варьироваться от 5 до 37, что явно нарушает требования Scrum (согласно гибкой методологии команда должна состоять из 5-7 разработчиков). Практически каждый проект SAP включает интеграцию с внешними системами. Вне зависимости от содержания и качества реализации спринтов проектной командой SAP, велика зависимость от внешнего ресурса. Игнорирование плана проекта SAP, задерживание тестирования и выпуска продукта, не информирование о внесённых изменениях и прочие действия внешней стороны критически влияют на выполнение спринта и не могут быть устранены в рамках Scrum.

Перейдём к функциональным особенностям. SAP-система представляет собой интегрированную среду, объединяющую данные, организационную структуру и процессы компании. Все три указанных звена являются общими и каждое из функциональных направлений может вносить свои изменения, поэтому спринты должны организовываться кросс-командно, что весьма нетривиально. Разумнее всего в число первых спринтов включить заведение организационной структуры, являющейся основополагающей частью предприятия. В последующих спринтах следует позаботиться о миграции и ведении основных и переменных данных для всех направлений бизнеса. И лишь потом сосредоточиться на реализации бизнес-процессов. Важно заметить, что всё вышеперечисленное ведётся в системе SAP настройками, за исключением, пожалуй, процессов, требующих значительной доработки. Тем самым, применение Scrum целесообразно только для реализации бизнес-процессов SAP, в то время как для организационной структуры и данных гибкая методология позволяет выполнить лишь демонстрацию и исправление очевидных ошибок.

И последнее, Agile Scrum анонсирует принцип регулярных поставок за счёт использования фич (Feature - переключатель). Тем самым реализованные программы переносятся в продуктивную среду по мере выполнения спринтов, включение и отключение внесённых изменений обрабатывается переключателями, чтобы не навредить существующим разработкам. К доработкам SAP вопросов нет, но как быть с настройками, ведь механизм переключателей здесь не работает? Более того, выполнив одну настройку, можно изменить другую, в модификации которой не было необходимости. Можно подумать о реализации дополнительных программ, выполняющих функции переключателей к настройкам, однако в SAP так много позиций кастомизации, что доработка выльется в неоправданно большие затраты.

**Таблица 2.** Особенности проекта SAP с точки зрения Scrum

| Особенность                        | Соответствие Agile Scrum | Комментарий                                                                |
|------------------------------------|--------------------------|----------------------------------------------------------------------------|
| Число участников проектной команды | Нет                      | Число участников много больше                                              |
| Интеграция с внешними системами    | Нет                      | Scrum не устраняет риски внешних сторон                                    |
| Данные                             | Нет                      | Быстрая доставка без возможности существенных изменений                    |
| Организационная структура          |                          |                                                                            |
| Бизнес-процессы                    | Да                       | Быстрая доставка с обратной связью                                         |
| Транспортные запросы               | Нет                      | Перенос в продуктивную систему только запросов разработки, но не настройки |

Подведём небольшой итог применения Scrum в больших проектах SAP, требующих как кастомизацию, так и разработку системы. Результаты обсуждения функциональных и организационных особенностей проекта SAP вынесены в табл. 2. Из данных таблицы можно сделать следующий вывод: значительное преимущество Scrum вносит только с точки зрения реализации бизнес-процессов, все прочие аспекты проекта существенно не улучшаются. Таким образом, использование Agile Scrum в проектах тиражирования и внедрения SAP «с нуля» выглядит весьма спорно (табл. 1., оценка целесообразности 1).

Вернёмся ещё раз к каскадной модели внедрения SAP. Возьмём манифест Agile [5] и попытаемся понять, насколько он покрывается водопадной моделью. Получим следующие результаты:

- принципы совместной работы команды и минимизации лишней работы отражены в модели водопад за счёт вовлечения ключевых пользователей и выполнении только включённых в объём проекта работ;
- принципы общения лицом к лицу и внимания к качеству находят частичное покрытие в каскадной схеме внедрения КИС;
- все прочие принципы и ценности Agile не релевантны в каскадной модели.

Разберёмся, почему в модели Waterfall применяется так мало принципов Agile. Возможно от того, что в каскадной схеме внедрения КИС акцент сделан на иные сущности. Проведём анализ типовых рисков проекта, идентифицированных Б. Бэмом, при усовершенствовании итерационной и создании спиралевидной модели [3]. Сгруппировав риски, выделим проблемные области имплементации КИС:

- нереальность сроков, бюджета, а также дефицит ресурсов проекта;
- технические недоработки и постоянные изменения;
- участие внешних сторон;
- низкая квалификация кадров.

Это позволяет сформулировать следующий вывод: метод Agile Scrum нацелен на решение технических проблем, возникающих в условиях непрекращающихся изменений, все прочие проблемы проекта вне Scrum. Таким образом, Agile Scrum хорош с точки зрения разработки программ, когда содержание продукта неизвестно ни проектной команде, ни пользователям. Однако применение Scrum требует глобальных изменений в организации проекта и не снижает прочих рисков неудачи.

При имплементации масштабных SAP решений, когда объём проекта понятен даже при отсутствии детализированных требований, технические неточности устраняются на этапах системного, интеграционного и приёмочного тестирования. Agile Scrum существенных преимуществ здесь дать не может, так же как и в случае, когда затраты на реорганизацию команды и следование Scrum значительно превосходят цикл разработки программного решения. Резюмируя вышесказанное, хочется отметить, что метод Scrum как разновидность итерационной модели ориентирован в большей степени на пользователей и разработчиков, в то время как каскадная схема – на руководителей проектов.

### Литература

1. Larman C., Basili V.R. Iterative and Incremental Development: A Brief History // Computer. – 2003. – Vol. 36, №6. – P. 47-56.
2. Основопологающие принципы Agile-манифеста [Электронный ресурс] // Agile Alliance – Режим доступа: <http://agilemanifesto.org/iso/ru/principles.html> (дата обращения 10.01.2018)
3. Гудков Е.А., Деревнина А.М., Катасонова Н.С. Анализ каскадной, итерационной и спиралевидной моделей внедрения корпоративных информационных систем // Корпоративные информационные системы. – 2018. – №1. – С. 16-27. – URL: <http://corpinfosys.ru/archive/issue-1/48-2018-1-models>
4. Стеллман Э., Грин Д. Постигая Agile. Ценности, принципы, методологии. – М.: Манн, Иванов и Фербер, 2018. – 448 с.

5. Применение SCRUM-подходов в проектах SAP [Электронный ресурс] // SapLand - Режим доступа: <https://sapland.ru/courses/for-order/primenenie-scrum-podhodov-v-proektah-sap.html> (дата обращения 10.01.2018)

### Выходные данные статьи

Степанов Д.Ю., Вельсовский А.В. Применение Agile Scrum в проектах SAP  
// Корпоративные информационные системы. - 2018. - №1. - С. 9-17. - URL:  
<http://corpinfosys.ru/archive/issue-1/46-2018-1-scrum>

### Об авторах



**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 публикации в журналах «Логистика сегодня», «САПтер» и др. Электронная почта: [mail@stepanovd.com](mailto:mail@stepanovd.com)



**Вельсовский Андрей Вельтерович** - руководитель группы разработки корпоративных информационных систем. Специализируется на среде программирования ABAP. Имеет более чем 15-летний опыт работы с решениями компании SAP, участвовал в многочисленных проектах по доработке систем. Электронная почта: [velsovskiyav@mail.ru](mailto:velsovskiyav@mail.ru)

## Анализ каскадной, итерационной и спиралевидной моделей внедрения корпоративных информационных систем

Гудков Евгений Алексеевич,  
Деревнина Ангелина Михайловна,  
Катасонова Наталья Сергеевна

**Аннотация:** в работе приводится описание каскадной, итерационной и спиралевидной моделей внедрения корпоративных информационных систем. Анализируются достоинства и недостатки каждой из моделей, рассматривается применимость моделей при имплементации сложных информационных систем.

Жизненный цикл (далее - ЖЦ) программного обеспечения (далее - ПО) состоит из ряда этапов, начинающихся стадией зарождения и заканчивающихся прекращением применения (рис. 1). Любая информационная система (далее - ИС) представляется совокупностью программных продуктов или ПО, тем самым определение жизненного цикла ПО и ИС тождественны. Вследствие того, что современные корпоративные информационные системы (далее - КИС) состоят из множества ИС, последнее применимо также и к КИС.



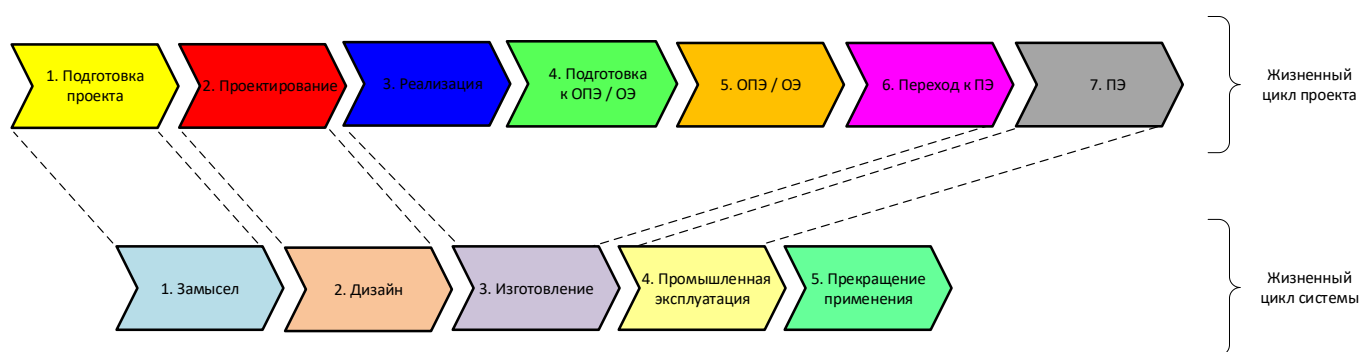
**Рис. 1.** Жизненный цикл программного обеспечения

Процесс внедрения КИС является составной частью ЖЦ. В работе [1] приводится типизация стадий имплементации КИС, включающая этапы подготовки, проектирования, реализации, опытно-промышленной и продуктивной эксплуатации. Этапы внедрения КИС задают последовательность операций, необходимых для успешного использования программного решения на предприятии заказчика. Тем самым можно говорить о двух жизненных циклах: ЖЦ корпоративной информационной системы и ЖЦ процесса её внедрения. Следуя данным рис. 2, этапы

циклов зачастую сопоставимы, однако акцент в первом случае сделан на программный продукт, а во втором – на ход его реализации. Часто процесс имплементации КИС называют моделью внедрения, которая задаёт порядок операций для имплементации системы, причём от модели к модели последовательность и содержание активностей разнятся. Выделяют три базовые модели внедрения КИС, все прочие рассматриваются как их производные [2]:

- каскадная;
- итерационная;
- спиралевидная.

Целью данной является анализ моделей внедрения корпоративных информационных систем для обеспечения эффективного процесса имплементации. Реализация цели потребует отдельного детального рассмотрения каждой из моделей.



**Рис. 2.** Сопоставление жизненного цикла системы и этапов внедрения КИС

## Каскадная модель внедрения корпоративных информационных систем

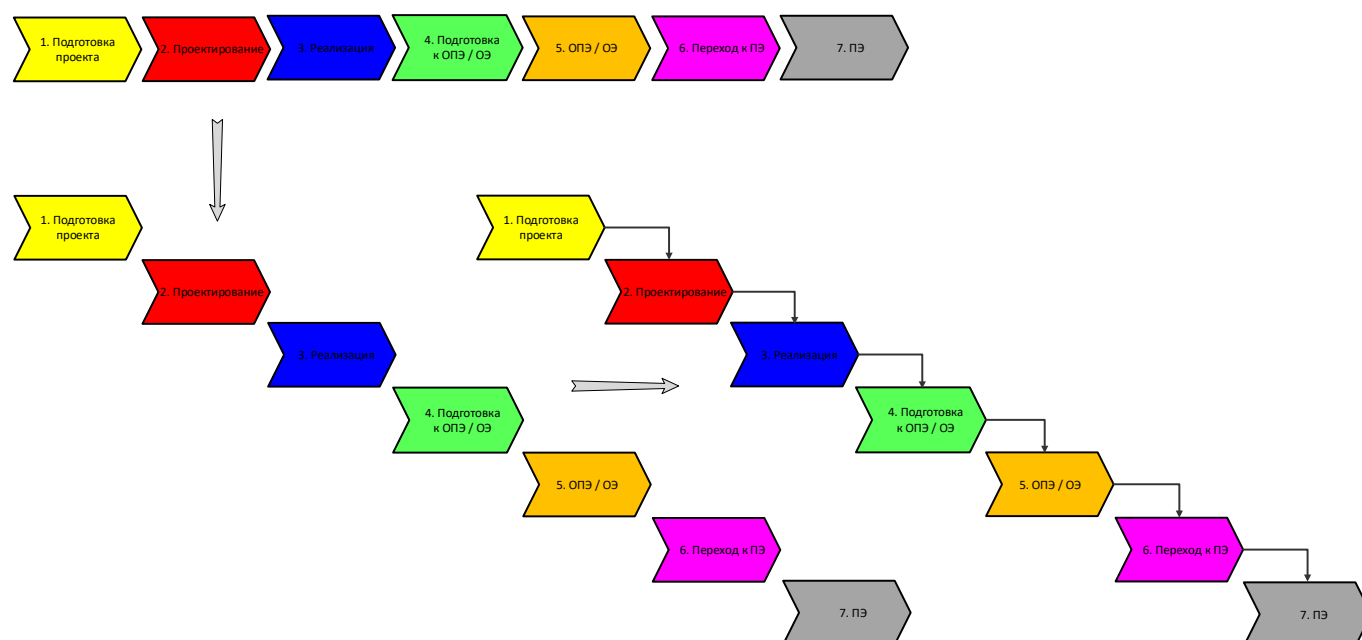
Воспользуемся типовыми этапами ЖЦ проекта внедрения (рис. 2). Преобразуем линейную последовательность следующим образом: каждый предыдущий этап сместим влево вверх, а каждый последующий – вправо вниз, тем самым получим схему, следующую слева направо и сверху вниз. Каскадная модель внедрения КИС образуется путём соединения полученных этапов между собой (рис. 3). Данная модель или, как её часто называют, модель водопад, была предложена в 1970 году У. Ройсом. Реализация проекта, согласно данной модели, ведётся путём строгого выполнения задач каждого из этапов (типовые этапы внедрения КИС), при этом переход к последующему этапу возможен лишь в случае успешного завершения предыдущей стадии [3]. Пропуск какого-либо из этапов, возврат к предыдущим стадиям и повторение этапов запрещены, именно по этой причине модель часто именуют

последовательной или однопроходной. Следуя рис. 3, очевидны достоинства и недостатки водопадной модели. К плюсам можно отнести:

- прозрачность определения сроков, работ и затрат;
- наличие согласованной процедуры перехода между этапами;
- независимость выполнения этапов,

минусами являются:

- невозможность устранения ошибок предыдущих этапов;
- отсутствие гибкости.



**Рис. 3.** Переход от типовых этапов к каскадной модели внедрения КИС

Проект внедрения КИС на основе данной модели состоит из активностей:

- подготовка проекта, заключающаяся в формировании основных концепций, стратегий и подходов к реализации функционала КИС;
- идентификация и анализ требований, предъявляемых к КИС и их приоритезация;
- формирование проектных решений и спецификаций, описывающих способ реализации ранее сформированных требований к КИС;
- кастомизация и доработка КИС на основе ранее подготовленных решений и спецификаций, описывающих реализацию требований;
- функционально-модульное, интеграционное и приёмочное тестирование выполненных настроек и доработок КИС;

- переход к продуктивной эксплуатации и поддержка.

Кроме того, каждый этап заканчивается валидацией и согласованием активностей следующей стадии, а также подтверждением возможности перехода к последующему этапу.

Вышеприведённый список демонстрирует закономерность: проектирование, реализация и тестирование КИС ведутся на основе требований, идентифицированных на ранних этапах. Тем самым, если изначальные требования были сформированы неверно, вся информационная система будет подготовлена некорректно (ранее отмеченный минус о невозможности устранения ошибок предыдущих этапов), что вероятнее всего потребует доделывания КИС. Отсутствие гибкости модели можно отнести как к минусу, так и плюсу: если ведётся частичное внедрение КИС незначительного по объёму функционала, возможные изменения требований по сравнению с начальными желательно включить в проект. Однако, когда имплементируется многофункциональная КИС, интегрированная с множеством как внешних, так и внутренних подсистем, незначительная корректировка требований может привести к значительным изменениям сроков, работ и затрат.

Именно поэтому каскадную модель часто применяют на проектах внедрения КИС «с нуля» и «тиражирования», когда объём выполняемых работ и их трудозатраты достигают внушительных величин. На подобных проектах возникает конфликт интересов между различными заинтересованными сторонами, поэтому результаты каждого из этапов тщательно документируются и согласуются во избежание разночтений и увеличения объёма работ. Невозможно запустить КИС в масштабах предприятия с учётом требований и пожеланий каждого из сотрудников. Поэтому объём задач фиксируется списком наиболее важных и подтверждённых требований ключевыми, но не конечными пользователями. В случае появления новых или изменения существующих требований предлагается регистрировать запрос на изменение. Суть последнего состоит в следующем: необходимые доработки КИС будут выполнены вне рамок проекта в отдельные сроки и бюджет.

### **Итерационная модель имплементации информационных систем**

Попытаемся устранить недостаток однопроходной модели. Для этого каждый из этапов схемы рис. 3 дополним контуром обратной связи, тем самым добавив возможность возврата на предыдущие стадии. Если внимательно проанализировать полученный результат, окажется, что каждый из этапов может выполняться несколько раз. Именно поэтому полученную модель (рис. 4) называют итерационной. Впервые

применённая в США ещё в 1957 году, многопроходная итерационная модель основана на концепции IID (Iterative and Incremental Development), включающей принципы итеративности (уточнение и детализация разрабатываемого программного обеспечения шаг за шагом), инкрементальности (увеличение функциональности ПО для каждой итерации) и эволюционности (максимальное использование наработок предыдущих итераций для последующих) [4].

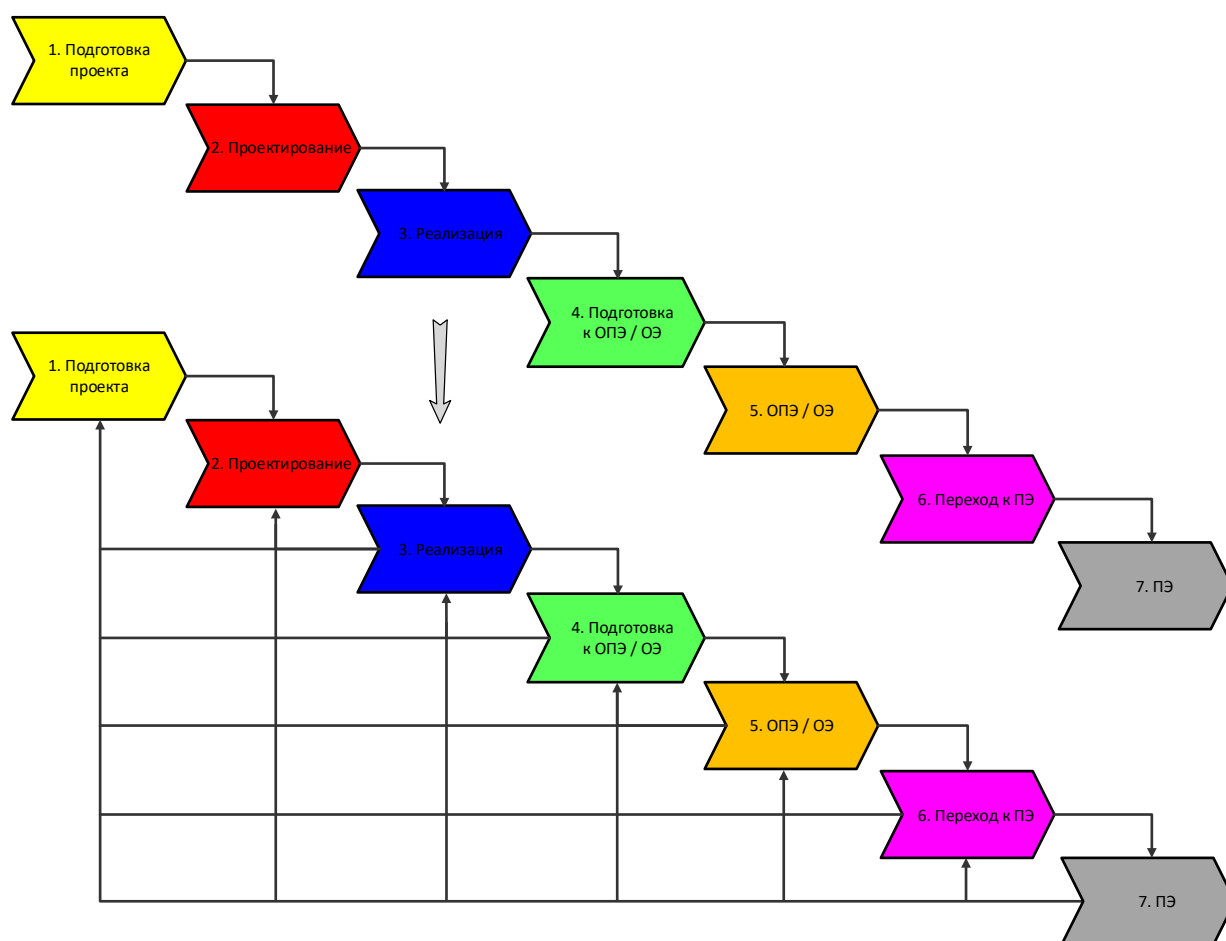


Рис. 4. Переход от каскадной к итерационной модели внедрения КИС

Разработка ПО согласно концепции IID сводится к разбиению этапа реализации на серию быстрых, лёгких и адаптивных итераций, оперативно приносящих результаты. Каждая итерация основана на PDCA-цикле Деминга (Plan-Do-Check-Act) и завершается демонстрацией потребителю полученного промежуточного продукта с целью скорейшего выявления потенциальных ошибок. Более того, в ходе выполнения итераций представление о конечном продукте изменяется, поэтому добавляются новые функциональные возможности. Продолжительность каждой итерации

варьируется в пределах 1-6 недель, а начальный список требований к ПО вообще может отсутствовать.

Отметим плюсы итерационной модели:

- оперативная разработка и демонстрация ПО для устранения ошибок;
- допускается отсутствие требований к ПО,

а также минусы:

- отсутствие понимания объёма работ для завершения проекта;
- ориентированность на разработку, но не кастомизацию ПО.

Рассмотрим проект имплементации КИС согласно предлагаемой модели:

- идентификация и анализ требований, предъявляемых к КИС, а также приоритезация найденных требований (опционально);
- определение числа и продолжительности итераций разработки КИС, в случае наличия приоритезированных требований их распределение по итерациям разработки;
- доработка и кастомизация КИС, функциональное и интеграционное тестирование с последующей демонстрацией полученного продукта заказчику для уточнения требований (для всех итераций);
- проведение приёмочного тестирования;
- документирование реализованного программного решения;
- переход к продуктивной эксплуатации и поддержка.

Выполнение итераций подразумевает демонстрацию продукта заказчику, в результате чего выявляются дефекты и идентифицируются новые требования к ПО. Первый минус многопроходной модели формулируется достаточно просто: вновь появляющиеся требования и пожелания клиента могут выйти за временные рамки изначально обсуждённых итераций. Таким образом, необходимо некое мерило требований для отсекаания маловажных, что в принципе противоречит самой концепции IID.

Второй минус выглядит куда более серьезно: раньше речь шла исключительно о доработке КИС, теперь же непонятно, как быть с кастомизацией и интеграцией. Сначала разберёмся с кастомизацией. Настройке в КИС подлежат уровни данных и бизнес-процессов: организационная структура, процессы, основные и переменные данные. Невозможно выполнить настройку, например, бизнес-процесса в КИС, следуя принципу итеративности: процесс или кастомизирован полностью, или не готов вовсе. Разумнее всего применить инкрементальный подход: разбить процесс на части

и настроить одну часть бизнес-процесса в заданной итерации, а остальные – в следующих. Данный подход применим также к оргструктуре и данным. В итоге получается некий аналог требований: процессы, организационная структура и данные должны быть декомпозированы на объекты и распределены по итерациям. Далее необходимо понять следующее: полученные объекты включать в начальные или завершающие итерации? Ответ связан с вопросом интеграции.

Интеграцию КИС можно разделить на внутреннюю и внешнюю. Под внутренней интеграцией подразумевается взаимодействие объектов и модулей КИС между собой, например, приход товара на склад должен порождать бухгалтерские проводки. В данном примере присутствуют две сущности различных модулей: приход и проводка, относящиеся к функциональности логистики закупки и финансам соответственно. Второй вид интеграции подразумевает взаимодействие КИС с внешними подсистемами, например, SRM, CRM, WMS и др. Если внутреннюю интеграцию в большей степени можно отнести к кастомизации КИС, то внешнюю – к настройке и доработке. Никакая доработка КИС не может вестись без наличия базовых компонентов системы, которые преимущественно заводятся путём настройки. Тем самым логично в начальные итерации включить кастомизацию КИС и интеграцию, и лишь затем доработку. Поэтому каждая итерация должна включать как функционально-модульное, так и интеграционное тестирование. Суммируя, видится следующая картина реализации итераций:

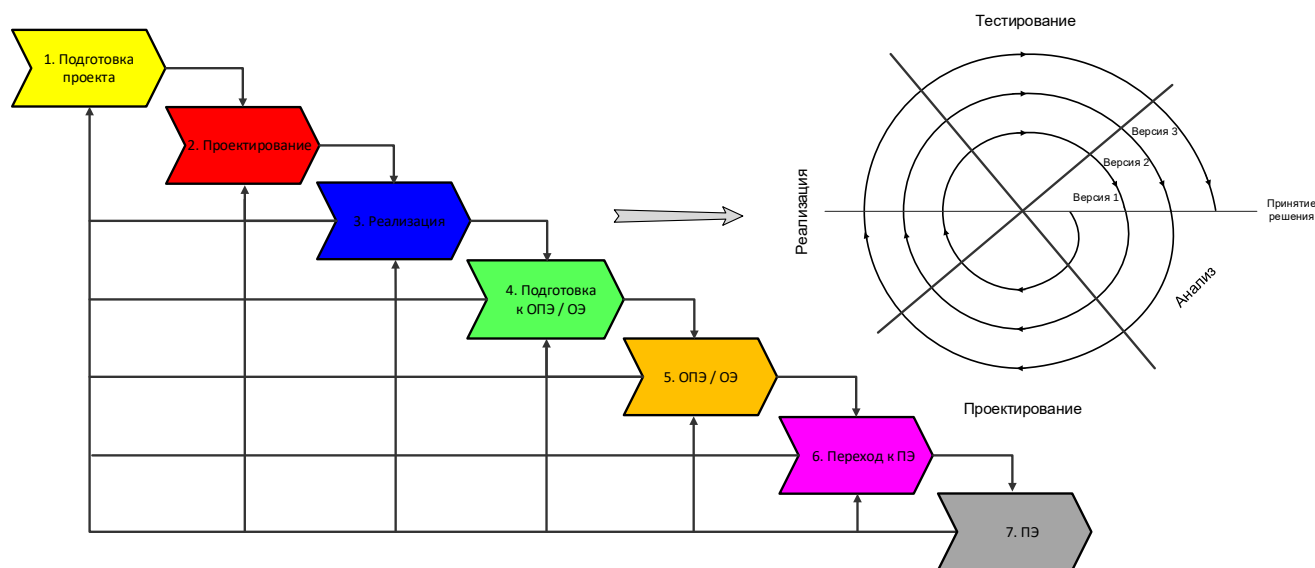
- начальные итерации должны включать настройку системы и внутреннюю интеграцию по инкрементальному принципу с целью подготовки основополагающего ядра КИС;
- последующие итерации содержат доработки КИС, использующие ранее подготовленные функции КИС путём кастомизации.

Резюмируя вышесказанное, применение итерационной модели вполне логично для доработки КИС, настройка же потребует дополнительных манипуляций. Несмотря на статистику [5], гласящую, что порядка 70% функционала иностранных КИС требуют доработки, пока многопроходная модель применяется в России достаточно редко. Возможно, причина кроется в том, что предпочтение отдаётся максимальному использованию стандартного функционала КИС, то есть кастомизации, против его доработки.

### **Спиралевидная модель внедрения КИС**

Разрешая минус итерационной модели, касающийся отсутствия понимания объёма проекта, в 1986 году Б. Бэмом была предложена спиралевидная модель

внедрения КИС (рис. 5). Данная схема сочетает в себе принципы как последовательной, так и многопроходной моделей. Основной акцент спиралевидной модели сделан на обработке 10 наиболее распространённых, по мнению Б. Бэма, рисков, (рис. 6) и оценивании степени готовности ПО [2].



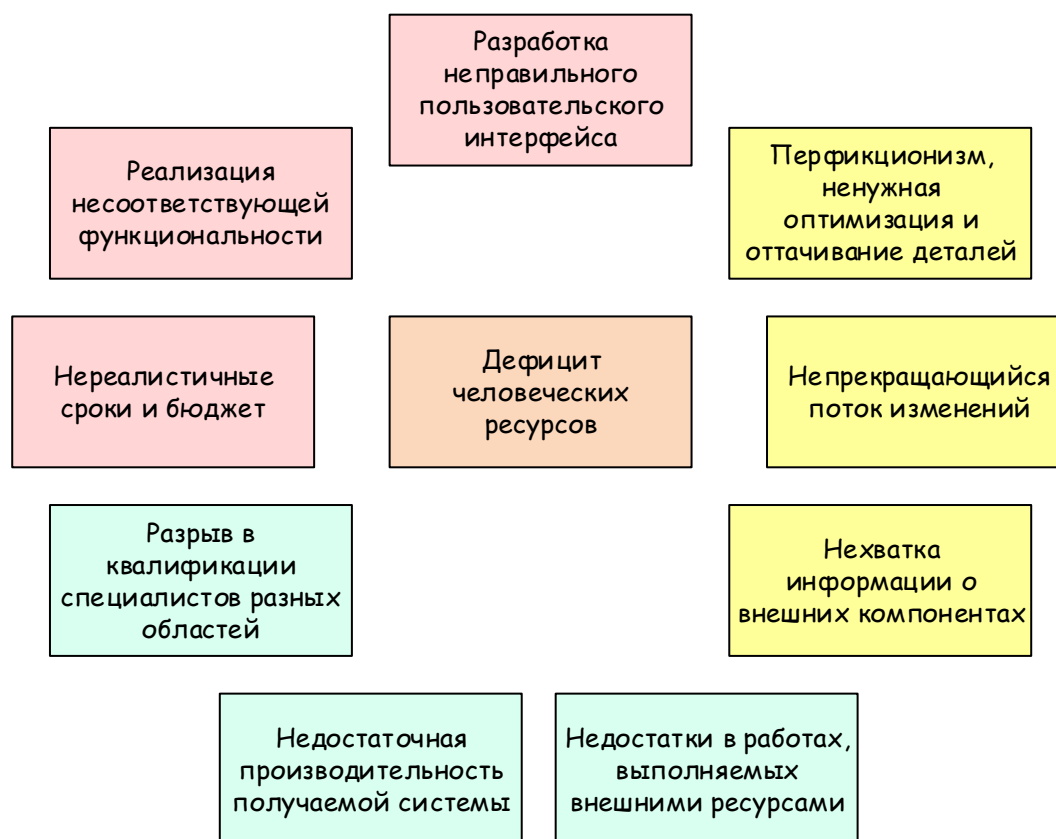
**Рис. 5.** Переход от итерационной к спиралевидной модели внедрения КИС

Следует отметить, спиралевидную модель нередко рассматривают как частный случай итерационной. Поэтому каждый виток спирали сравним с итерацией одноимённой модели внедрения КИС и характеризуется:

- определением цели и задач;
- обработкой рисков;
- разработкой, настройкой и тестированием ПО;
- оцениванием результатов и планированием следующего витка разработки.

Проект имплементации КИС на основе спиралевидной модели сопоставим с итерационной с той лишь разницей, что каждый виток спирали (в контексте многопроходной модели виток – это итерация) помимо обработки рисков должен включать:

- проверку неперевышения сроков и бюджета проекта;
- оценку необходимости выполнения ещё одного витка спирали;
- оценку уровня понимания требований к системе;
- анализ целесообразности завершения проекта.



**Рис. 6.** Типовые риски спиралевидной модели при внедрении КИС

Обработка всевозможных параметров проекта (сроки, ресурсы, бюджет, качество, коммуникации и др.), а также оценивание необходимости выполнения следующих витков разработки и завершения проекта требуют значительных трудовых затрат. Именно поэтому применение данной модели целесообразно в больших проектах внедрения КИС. Для небольших проектов или частичного внедрения КИС время, потраченное на обработку и оценивание параметров проекта, будет несопоставимо больше самого процесса разработки системы, что делает применение спиралевидной схемы нецелесообразным.

## Заключение

Несмотря на различие в порядке выполнения задач той или иной модели имплементации КИС, их содержание достаточно схоже: анализ требований, проектирование, разработка и тестирование. Если обратить внимание на итерационную и каскадную модели с точки зрения реализации проекта, можно заметить, что они уточняют и дополняют каскадную схему: стадии приёмочного тестирования, внедрения и поддержки в данных схемах идентичны однопроходной модели.

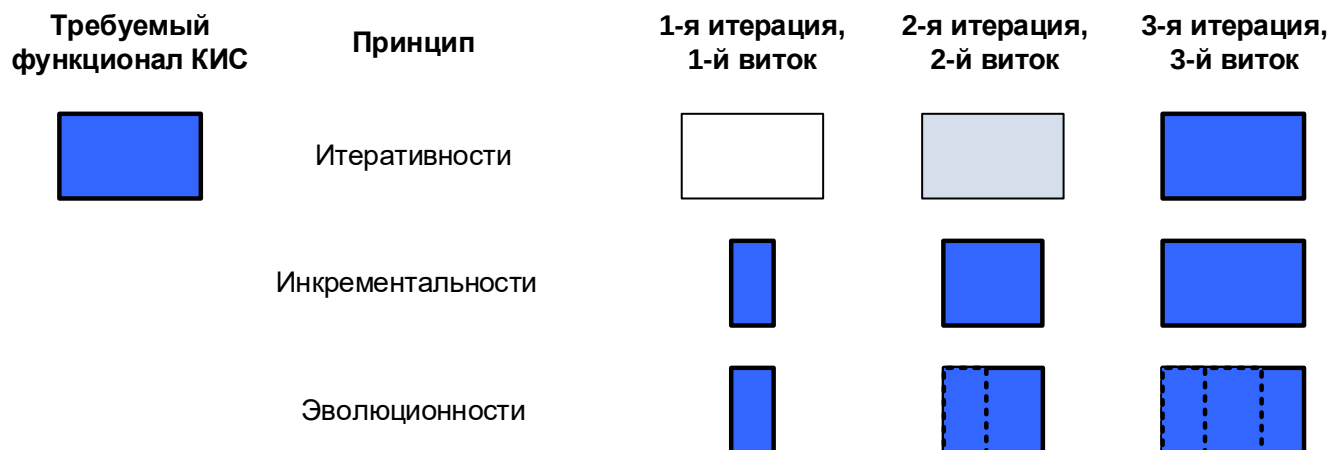


Рис. 7. Базовые принципы итерационной и спиралевидной моделей внедрения ПО

Фактически можно выделить лишь две модели внедрения КИС: однопроходная и многопроходная. Первая схема организована таким образом, что разработчик и функциональный консультант являются лицом, принимающим решение по форме, содержанию и особенностям требуемого ПО. Мнение ключевого или конечного пользователя относительно программы поступают настолько поздно, что становится невозможным кардинально изменить уже подготовленный программный продукт. Более того, однопроходная модель внедрения жёстко регламентирована с точки зрения документооборота и порядка выполнения работ, в частности, согласно РМВоК [6].

Напротив, в многопроходной схеме конечное содержание программного продукта определяется пользователем, но не разработчиком. Если многопроходная схема включает чётко выраженный уровень проекта, отвечающий за управление ресурсами, сроками, рисками и прочими стратами, говорят о спиралевидной модели внедрения КИС, в противном случае – итерационной. Обе схемы базируются на принципах IID, наглядно продемонстрированных на рис. 7.

Каскадная и спиралевидная модели преимущественно используются в больших проектах внедрения КИС, требующих тиражирования или внедрения решения «с нуля». Несмотря на наличие очевидных плюсов многопроходной схемы, на практике преимущественно используется однопроходная модель, адаптированная как к разработке, так и кастомизации функционала КИС. Область применения итерационной схемы видится в доработке существующих функциональных возможностей КИС.

Кроме описанных моделей внедрения существует огромное число прочих схем, организованных на основе их базовых принципов. В частности, DSDM (Dynamic Systems Development Method – метод разработки динамических систем), XP (eXtreme Programming – экстремальное программирование), Scrum, FDD (Feature Driven Development – разработка, управляемая функциональностью) и другие. Подобные методы объединены в рамках методологии Agile [7], подразумевающей гибкую разработку программ [4]. Последняя является перспективной областью дальнейших исследований авторов.

### Литература

1. Степанов Д.Ю. Обзор проектных документов при внедрении корпоративных информационных систем // Вопросы экономических наук. – 2014. – Т. 70, №6. – С. 54-62.
2. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем: учебное пособие. – Ростов н/Д.: Феникс, 2009. – 508 с.
3. Зараменских Е.П. Управление жизненным циклом информационных систем. – М.: Юрайт, 2017. – 431 с.
4. Larman C., Basili V.R. Iterative and Incremental Development: A Brief History // Computer. – 2003. – Vol. 36, №6. – P. 47-56.
5. Левочкина Г., Калянов Г., Васильев Р. Управление развитием информационных систем. – М.: Горячая Линия – Телеком, 2014. – 376 с.
6. Руководство к своду знаний по управлению проектами (Руководство РМВОК). Пятое издание. – М.: Олимп-Бизнес, 2014. – 586 с.
7. Стеллман Э., Грин Д. Постигая Agile. Ценности, принципы, методологии. – М.: Манн, Иванов и Фербер, 2018. – 448 с.

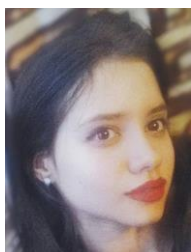
### Выходные данные статьи

Гудков Е.А., Деревнина А.М., Катасонова Н.С. Анализ каскадной, итерационной и спиралевидной моделей внедрения корпоративных информационных систем // Корпоративные информационные системы. – 2018. – №1. – С. 18-29. – URL: <http://corpinfosys.ru/archive/issue-1/48-2018-1-models>

### **Об авторах**



**Гудков Евгений Алексеевич** - студент 4-го курса кафедры оптических и биотехнических систем и технологий МТУ МИРЭА. Тема выпускной квалификационной работы бакалавра «Применение спиралевидной модели внедрения информационных систем в городской поликлинике». Электронная почта: eh1770802@gmail.com



**Деревнина Ангелина Михайловна** - студентка 4-го курса кафедры оптических и биотехнических систем и технологий МТУ МИРЭА. Тема выпускной квалификационной работы бакалавра «Разработка автоматизированного рабочего места врача-невролога с использованием метода Agile Scrum». Электронная почта: linarusty@yandex.ru



**Катасонова Наталья Сергеевна** - студентка 4-го курса кафедры оптических и биотехнических систем и технологий МТУ МИРЭА. Тема выпускной квалификационной работы бакалавра «Автоматизация ключевых бизнес-процессов городской больницы на основе каскадной модели внедрения». Электронная почта: natasha0596@mail.ru

## Особенности проекта внедрения MRP по точке перезаказа

Степанов Дмитрий Юрьевич

**Аннотация:** статья содержит описание сложностей, возникающих при внедрении функционала планирования потребности в материалах на основе точки перезаказа (Reorder Point). Разбираются вопросы сбора статистической информации, интеграции бизнес-процессов, обработки и загрузки данных в корпоративную информационную систему, а также постепенного продуктивного запуска.

Внедрение корпоративных информационных систем (далее – КИС) имеет под собой вполне резонное обоснование: автоматизация бизнес-процессов, позволяющая сосредоточить внимание сотрудников на наиболее важных операциях кампании, минимизируя рутинные малозначимые транзакции. Имплементация КИС может вестись на основе стратегий полного или лоскутно-кусочного внедрения. В последнем случае лишь часть ключевых процессов предприятия подлежат покрытию функционалом КИС.

Одним из немногих функционалов информационной системы, выносимых в отдельный подпроект внедрения, является планирование потребностей в материалах (Material Requirement Planning, далее – MRP) [1]. Существуют различные типы MRP в зависимости от вида производства и сложности планирования: планирование на основе потребления, планирование по точке перезаказа (Reorder Point, далее – ROP), сезонное планирование и др.

Простейшим видом MRP является планирование по точке перезаказа (Reorder Point – ROP). Суть ROP сводится к измерению параметров, характеризующих состояние склада: текущий уровень запаса продукции и значение точки перезаказа для неё. Если значение точки перезаказа превышает текущий уровень запаса, запускается процедура пополнения продукции за счёт внутреннего производства или закупки у внешнего поставщика (рис. 1). Внедрение ROP в стандарте ERP (Enterprise Resource Planning) пророчит сложности [2]. В частности:

- сбор статистической информации;
- интеграция бизнес-процессов и управление изменениями КИС;
- обработка и загрузка данных в КИС;
- запуск планирования материалов в КИС.

Сбор статистики, обработка и загрузка данных, а также постепенный запуск планирования присущи любому проекту вне зависимости от момента имплементации процедуры ROP, в то время как выравнивание существующих процессов для использования точки перезаказа – горькая неизбежность, если данный функционал внедряется много позже базового ERP. Рассмотрим далее каждый из пунктов подробнее.

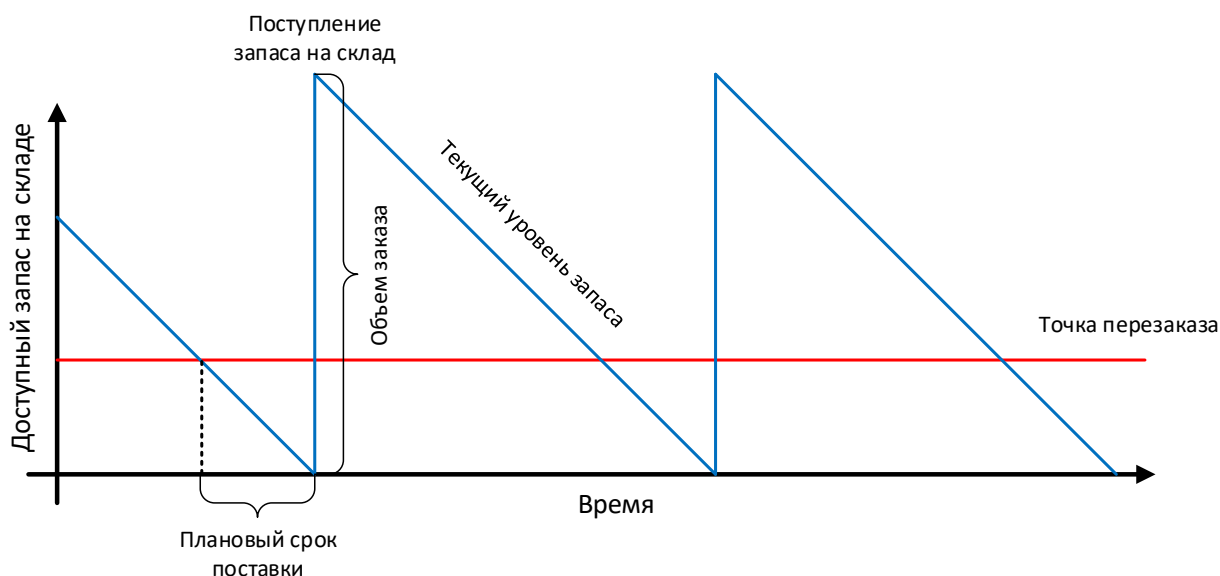


Рис. 1. Принципиальная схема планирования по точке заказа

### Сбор статистической информации

Схематически процедура ROP дана на рис. 1. Анализируя приведённый рисунок, можно выделить следующие основные параметры, используемые при обработке точки перезаказа:

- значение текущего запаса продукции на складе;
- значение точки перезаказа для продукции;
- время пополнения запаса (плановые сроки поставки);
- объём заказа.

Рассмотрим, как работает процедура ROP в КИС. Ежедневно, чаще всего в ночные часы, на складе выполняется замер текущего уровня запаса продукции. Предварительно вручную для каждой номенклатуры продукции устанавливается значение точки перезаказа. Если значение запаса ниже точки перезаказа, запускается процедура пополнения, допустим через закупку от поставщика.

В ходе закупки продукции, её запас должен быть положительным до момента оприходования на склад. Таким образом, после наступления события перезаказа и фактическим пополнением складской запас должен быть не нулевым, во избежание дефицита продукции и остановки работы компании. В формуле 1 приведена связь параметров, характеризующих процесс пополнения запасов.

$$\text{ТочкаПерезаказа} = \text{ПотреблениеВДень} / \text{ПлановыеСрокиПоставки}, \quad (1)$$

где ПотреблениеВДень характеризует объём потребляемой продукции за 1 день, ПлановыеСрокиПоставки – сроки поставки материалов от поставщика на склад в днях. Итого, если ежедневно требуется 5 штук продукции, а время поставки составляет 3 дня, то точка перезаказа согласно (1) будет равна 15. Расчёт вполне интуитивен: если каждый день потребляется 5 единиц материала, то имеется 3 дня ( $15/5=3$ ) для пополнения склада, что и будет являться плановым сроком поставки.

Запомним параметры формулы (1) для дальнейшего анализа. Рассмотрим следующий параметр из рис. 1: объём заказа. Как же правильно рассчитать данный параметр? Воспользуемся формулой Уилсона для определения оптимальной партии поставки (Economic Order Quantity, далее – EOQ) [3].

$$\text{EOQ} = \sqrt{(2 * \text{ЗатрОбрЗак} * \text{ПотребВМес} / \text{ЗатратХранСклад})}, \quad (2)$$

где ЗатрОбрЗак определяют затраты в рублях на обработку 1 заказа на поставку продукции от поставщика до склада, ПотребВМес – количество продукции, потребляемое за 1 месяц, ЗатратХранСклад – затраты в рублях на хранение данной продукции на собственном складе в месяц. Последний параметр зависит от стоимости продукции и рассчитывается по формуле (3).

$$\text{ЗатратХранСклад} = \text{Процент} * \text{СтоимостьПродукции}, \quad (3)$$

где СтоимостьПродукции задаёт цену закупаемой продукции, а Процент определяет % на затраты хранения в зависимости от стоимости.

Так что же дает EOQ? Формулы (2)-(3) задают количество продукции, которое будет заказываться у поставщика каждый раз при срабатывании точки перезаказа. В связи с этим возникает закономерный вопрос, а какое количество продукции оптимально заказывать и каково оптимальное число заказов у поставщика за месяц?

Именно на эти вопросы даёт ответ формула (2): чем выше стоимость обработки одной закупки от поставщика по сравнению с затратами на хранение продукции на собственном складе, тем меньше заказов должно быть создано, то есть каждый заказ

будет обеспечивать поставку на большее количество продукции в виду дешевизны хранения на складе. Обратное тоже верно.

Подводя итог, список параметров, необходимый для расчёта ROP, данный в начале раздела, следует уточнить и записать в следующем виде:

- значение текущего запаса продукции на складе;
- значение потребления продукции в день и за 1 месяц;
- время пополнения запаса (плановые сроки поставки);
- затраты на обработку 1 заказа на поставку;
- стоимость продукции.

Таким образом, выполнение простейшей задачи по определению значения точки перезаказа превращается в сбор и анализ значительного объёма исторической информации, требующий немалых трудозатрат. Так может не тратить время и указать значение данных параметров, не особо вдаваясь в детали? Согласно формуле (1) подобное может привести к дефициту продукции, когда запас материала полностью исчерпан, а его пополнение ещё не выполнено; следуя формулам (2)-(3), неизбежно увеличение себестоимости продукции за счёт косвенных затрат.

### **Интеграция бизнес-процессов и управление изменениями КИС**

Имплементация процедуры ROP в случае, когда базовые бизнес-процессы уже реализованы в КИС, требует выравнивания бизнес-транзакций системы. Рассматривая функционал точки перезаказа современных КИС, следует упомянуть о ABC и XYZ анализе [4], характеризующем стоимость запасов и регулярность их потребления (рис. 2). В проектах внедрения КИС, например, SAP ERP [5], функционал ROP используется для уровня CX матрицы ABC-XYZ. В частности, широкое применение точки перезаказа нашло в планировании запасных частей на производственных предприятиях. Обычно стоимость запчастей невысока, а регулярность потребления достаточно стабильна, тем самым подобные материалы отождествимы с CX уровнем матрицы.

Однако даже в этом случае требуется рассмотрение непрерывных интегрированных процессов: запасные части необходимо закупить, оприходовать на склад для дальнейшего потребления, и, возможно, перепродать заказчику. Тем самым, введение простейшей процедуры планирования ROP требует сопоставления как минимум 3 процессов. Кроме того, новейшие КИС позволяют осуществлять как закупку, так и сбыт под конкретную потребность, говоря простым языком, адресно, что так же должно найти отражение в процедуре планирования ROP. Так что же нужно

делать при внедрении ROP с точки зрения интеграции процессов? Воспользуемся схемой идентификации изменений (рис. 3):

- во-первых, требуется понять, на какие существующие процессы функционал ROP явно влияет. Например, как было сказано ранее, операции ремонта связаны с закупкой, запасами и сбытом. В результате проделанного анализа может возникнуть необходимость в кастомизации процессов КИС, ранее не планируемых к изменению;

|                                                          |                                                            |                                                              |
|----------------------------------------------------------|------------------------------------------------------------|--------------------------------------------------------------|
| <b>AX</b><br>Высокая стоимость<br>Регулярное потребление | <b>AY</b><br>Высокая стоимость<br>Нестабильное потребление | <b>AZ</b><br>Высокая стоимость<br>Стохастическое потребление |
| <b>BX</b><br>Средняя стоимость<br>Регулярное потребление | <b>BY</b><br>Средняя стоимость<br>Нестабильное потребление | <b>BZ</b><br>Средняя стоимость<br>Стохастическое потребление |
| <b>CX</b><br>Низкая стоимость<br>Регулярное потребление  | <b>CY</b><br>Низкая стоимость<br>Нестабильное потребление  | <b>CZ</b><br>Низкая стоимость<br>Стохастическое потребление  |

Рис. 2. ABC и XYZ матрица стоимости запасов и регулярности потребления

- во-вторых, путём регрессионного тестирования можно идентифицировать существующие процессы КИС, выполнение которых в заданной последовательности потенциально влияет на работу процедуры ROP. В подобных случаях порядок выполнения бизнес-транзакций подлежит изменению. Например, открытые заказы на закупку рассматриваются SAP ERP как плановые поступления, что снижает объём закупаемой продукции согласно ROP. Последнее приведёт к дефициту товаров в случае, если заказы на закупку открыты, однако срок действия их давно истёк. Изменение процесса потребует регулярного контроля и закрытия подобных заказов;
- и, наконец, в-третьих, процедура планирования сильно зависит от данных системы как основных, так и переменных. Малейшая ошибка, например, в основной записи материала, может привести к совершенно непоправимым результатам планирования. Как было рассмотрено на примере выше, аналогично дело обстоит и с переменными данными. Если до внедрения функционала

точки перезаказа пользователям КИС позволялась некая неточность ведения данных, то после – исключено.

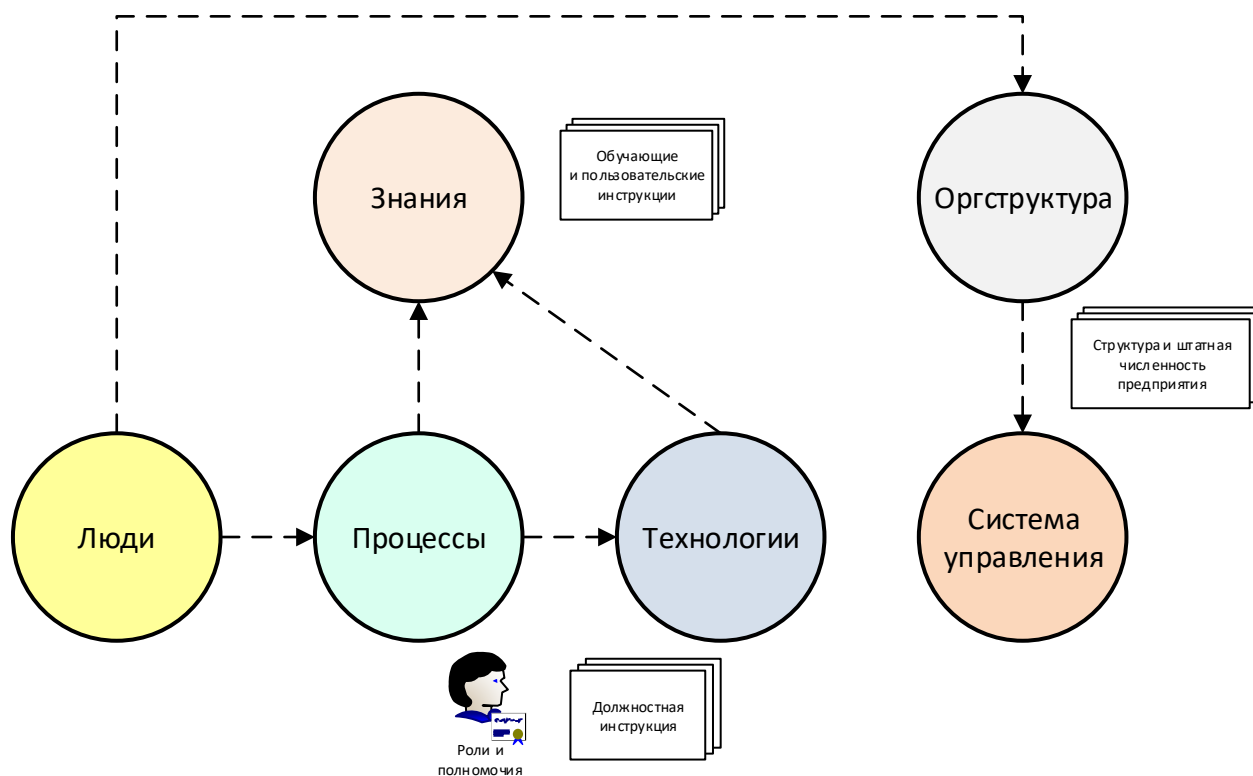


Рис. 3. Схема идентификации и управления изменениями

### Обработка и загрузка данных в КИС

Выполнив сбор, анализ и расчёт параметров планирования материалов переходим к не менее важному этапу: обработке и загрузке данных в КИС. Типовые шаги обработки данных приведены на рис. 4 и включают: очистку данных, в ходе которой удаляются устаревшие и некорректные данные, потенциально влияющие на ход работы информационной системы; выгрузку информации во внешний файл для целей дальнейшего анализа и последующей трансформации, суть последней заключается в преобразовании данных к формату, необходимому для загрузки в КИС; а также непосредственную загрузку в систему [6]. Важно отметить, что каждая из перечисленных активностей завершается операцией проверки результатов, валидация позволяет устранить человеческие ошибки в случае очистки, выгрузки и трансформации, кроме того, убедиться в правильности финальной загрузки данных в корпоративную информационную систему.

Рассмотрим применение вышеуказанного подхода к обработке данных для случая планирования по точке перезаказа. Суммируя данные рис. 4, выделим следующую последовательность для подготовки данных планирования:

- ответственному за планирование сотруднику поступает запрос на включение нового материала в список планирования по ROP;
- сотрудник планирования собирает и анализирует параметры планирования для нового материала, используя историческую информацию. В результате определяются: значение точки перезаказа, плановые сроки поставки и объём заказа;

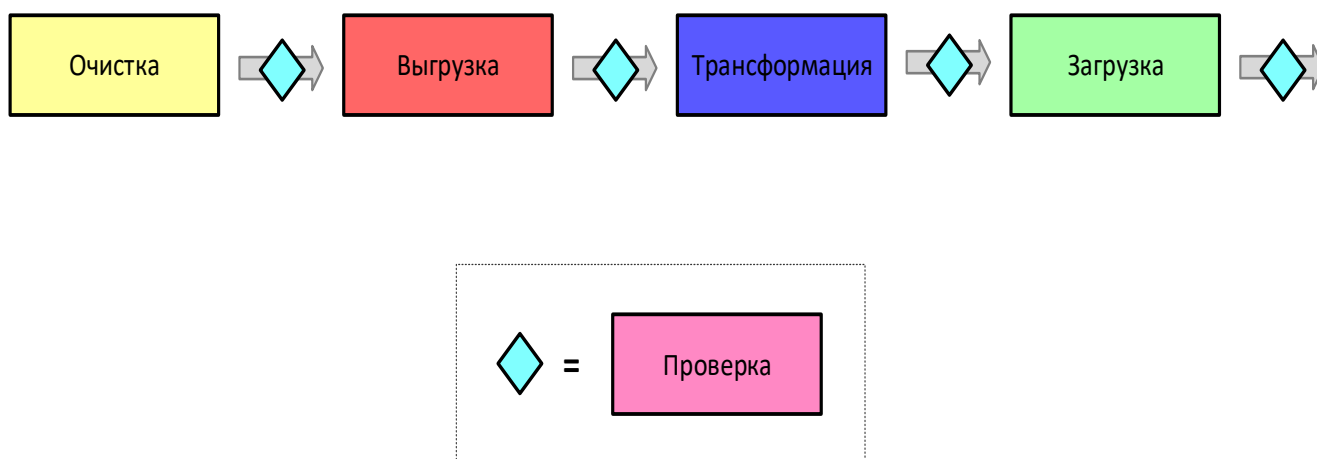


Рис. 4. Процесс обработки и загрузки данных в КИС

- ответственный за данные сотрудник проводит их очистку в КИС исключительно в разрезе нового материала. Очистке подлежат переменные данные системы, как то: заявки и заказы на закупку, резервирования, сбытовые заказы, сетевые графики проекта и прочие объекты на примере системы SAP ERP;
- сотрудник планирования или ответственный за данные осуществляет выгрузку мастер-данных во внешний файл с целью дальнейшего обновления и трансформации. Основными данными системы SAP ERP могут служить материалы, инфо-записи, книги источников поставок и прочее;
- используя ранее найденные параметры планирования, планировщик преобразует результаты выгрузки в файл трансформации, имеющий заданный формат. Файл передаётся команде, ответственной за данные, для дальнейшего ввода в КИС;
- загруженные в КИС данные подлежат валидации как сотрудником, отвечающим за данные, так и планировщиком.

Указанная последовательность потенциально может быть упрощена. Для этого рекомендуется ввести регулярную процедуру очистки как основных, так и переменных данных. В любом случае такая потребность возникнет, вопрос только времени. Ведь уже неоднократно подчёркивалось, что планирование материалов очень чувствительно к качеству ведения данных информационной системы.

### **Запуск планирования материалов в КИС**

Хорошей практикой считается продуктивный запуск КИС с использованием подэтапа стабилизации (рис. 5) [7]. Под стабилизацией промышленного запуска подразумевается последовательное увеличение нагрузки, подаваемой на информационную систему. После успешного прохождения ряда итераций, увеличивающих объём обрабатываемых основных и переменных данных КИС, следует переход к полноценному использованию системы. В чём смысл данного подхода? Основная идея заключается в быстром устранении недочётов и дефектов КИС на ранних этапах эксплуатации, когда объём данных ещё минимален. Более того, конечные пользователи проходят проверку «боем», допуская, обсуждая и устраняя ошибки при тщательной поддержке, не нанося значительного урона системе.

Если обратиться к функционалу ROP, стабилизация работы системы будет заключаться в последовательном увеличении числа материалов, планируемых по точке перезаказа. Допустим, в первую неделю планируется 100, далее 500 и, наконец, 1000 материалов. Планирование на минимальном объёме данных позволяет выявить возможные недоработки как информационной системы, так и бизнес-процессов. Пользователь, курирующий планирование, должен убедиться в корректности следующих позиций:

- обновлённых данных;
- функционала планирования ROP;
- созданных ROP документов.

Действительно, после внедрения процедуры планирования обновление данных становится процессом «бизнес как есть» (Business As Usual), минуя разовые активности перехода (Cutover Activities). Сама процедура обновления подразумевает подготовку списка материалов для планирования, сбор статистической информации и расчёт параметров планирования, очистку исторических данных, а также загрузку и валидацию данных планирования, следуя подходу на рис. 4.

Кроме того, запускается фоновая задача по планированию материалов и отслеживаются результаты её работы. В случае выявления некорректно созданных документов после запуска ROP, идентифицируется причина и устраняются последствия. Чаще всего причинами ошибочно созданных документов являются несовершенство введения данных или ошибки пользователей при реализации процессов в КИС. Выполнив несколько раз упомянутые активности, ответственные пользователи получают опыт, позволяющий обрабатывать большие объёмы данных при последующем увеличении числа планируемых по ROP материалов.

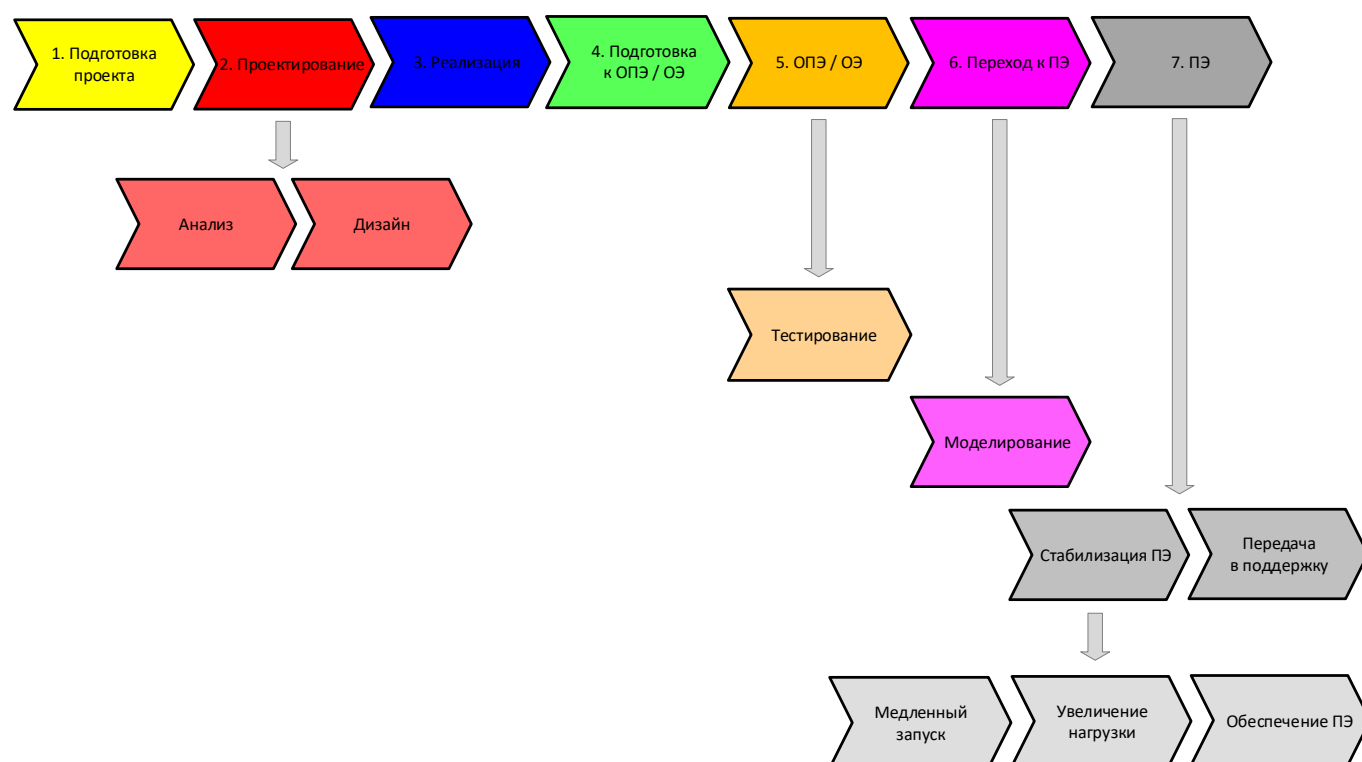


Рис. 5. Типовая методология внедрения КИС

## Заключение

В работе были рассмотрены типовые сложности, возникающие в процессе внедрения функционала планирования потребностей в материалах на основе точки перезаказа. Достаточно сложно описать все возможные проблемы имплементации в виду сильной зависимости от выбранного программного продукта КИС, как то: SAP ERP, Oracle eOBS, MS NAV или др. Цель же данной работы заключалась в задании тех проблемных областей, с которыми неминуемо столкновение в процессе внедрения. Перспективным направлением дальнейшего развития работы является анализ

планирования материалов на основе потребления и разузлования промышленных спецификаций.

### **Литература**

1. Гаврилов Д.А. Управление производством на базе стандарта MRPII. – СПб.: Питер, 2008. – 416 с.
2. Wallace T. ERP: Making it happen. – Canada: John Wiley & Sons, 2001. – 375 p.
3. Mandal B. Decision making in Inventory Management. – NY.: LAP Lambert Academic Publishing, 2011. – 118 p.
4. Логистика / Дыбская В.В. и др. – М.: Эксмо, 2009. – 944 с.
5. SAP ERP. Построение эффективной системы управления. / Пер. с англ.; под ред. Т. Качинской. – М.: Альпина Паблишер, 2008. – 356 с.
6. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: уровень данных. – М., 2017. – URL: [http://stepanovd.com/training\\_erp\\_1-10ru.html?lang=RU](http://stepanovd.com/training_erp_1-10ru.html?lang=RU).
7. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: модели и уровни внедрения систем. – М., 2017. – URL: [http://stepanovd.com/training\\_erp\\_1-3ru.html?lang=RU](http://stepanovd.com/training_erp_1-3ru.html?lang=RU).

### **Выходные данные статьи**

Степанов Д.Ю. Особенности внедрения MRP по точке перезаказа  
// Корпоративные информационные системы. – 2018. – №1. – С. 30-39. – URL:  
<http://corpinfosys.ru/archive/issue-1/47-2018-1-rop>

### **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 журналах «Логистика сегодня», «Вопросы экономических наук» и др. Электронная почта: [mail@stepanovd.com](mailto:mail@stepanovd.com)

## Сложности внедрения WMS-систем на примере SAP ERP

Терентьев Илья Михайлович

**Аннотация:** в статье приводятся типовые сложности, возникающие при внедрении WMS-систем: создание организационной структуры, миграция остатков при переходе к SAP ERP, перезикетирование, обучение пользователей, использование карманных ПТК и печать этикеток. Работа будет полезна консультантам и аналитикам, занимающимся модулем SAP WM.

В процессе внедрения модуля WM (Warehouse Management – система управления складами) системы SAP ERP может возникнуть множество вопросов, особенно если имплементация подразумевает использование функциональности единиц обработки (HU, Handling unit). Рассмотрим лишь некоторые из них: создание организационной структуры, миграция остатков при переходе к SAP ERP, перезикетирование, обучение пользователей, использование карманных ПТК и печать этикеток.

### Создание организационной структуры

При создании организационной структуры (рис. 1) часто возникает такая ситуация, когда ключевые пользователи постоянно меняют своё мнение по поводу того, как отразить реальную структуру в системе SAP ERP. Например, несколько помещений, комнат и даже зданий можно объединить в одну структурную единицу в SAP или наоборот из одного помещения сделать несколько структурных единиц. В качестве одного из вариантов решения является применение карты склада/складов, где можно обрисовать текущую схему складов и наложить на неё сетку структурных складов системы SAP. Это позволит ключевым пользователям совместно с функциональными консультантами ещё раз оценить правильность создания организационной структуры в SAP ERP.

Для того, чтобы понять складскую структуру, функциональному консультанту очень поможет экскурсия по складским помещениям. Обычно одной экскурсии бывает недостаточно. Рекомендуется следующий подход: организовать посещение склада, получить карту склада от ключевых пользователей, далее наложить на неё сетку организационной структуры SAP ERP, затем ещё раз посетить склад и лишь потом финализировать структуру склада в SAP.

Дополнительно к этому может возникнуть вопрос определения складских мест. Чаще всего формат задания складских мест (название, расположение и др.) уже

имеется, даже если нет никакой автоматизированной информационной системы. Важно проверить соотносится ли формат складских мест с системой SAP.

|     |     |     |
|-----|-----|-----|
| A11 | A12 | A13 |
| A21 | A22 | A23 |
| A31 | A32 | A33 |
| A41 | A42 | A43 |
| A51 | A52 | A53 |
| A61 | A62 | A63 |

|      |      |      |      |
|------|------|------|------|
| B11  | B12  | B13  | B14  |
| B21  | B22  | B23  | B24  |
| B31  | B32  | B33  | B34  |
| B41  | B42  | B43  | B44  |
| B51  | B52  | B53  | B54  |
| B61  | B62  | B63  | B64  |
| B71  | B72  | B73  | B74  |
| B81  | B82  | B83  | B84  |
| B91  | B92  | B93  | B94  |
| B101 | B102 | B103 | B104 |
| B111 | B112 | B113 | B114 |
| B121 | B122 | B123 | B124 |

|    |    |    |    |
|----|----|----|----|
| C1 | C2 | C3 | C4 |
| D1 | D2 | D3 | D4 |
| E1 | E2 | E3 | E4 |

Рис. 1. Пример организационной структуры склада

## Миграция остатков при переходе от NSA систем к SAP ERP

При переходе от NSA систем (NSA, Non Sap Application - не SAP-системы) к SAP ERP могут возникнуть проблемы миграции складских остатков. Обычно на каждом предприятии есть как минимум система 1С для ведения учёта запасов и, возможно, дополнительная система складского учёта. Основными вопросы, на которые следует обратить внимание при переходе, являются:

- Несоответствие между системами, например, в 1С есть остаток материала, а в складской системе нет или наоборот. Подобное возникает в связи с тем, что какие-то операции забыли или ещё не успели отразить в системе. Важно изначально определить, какая из систем будет являться главной (мастер-система), именно в ней по умолчанию будут находиться верные остатки.
- Приведение систем в соответствие с оргструктурой SAP ERP для последующего соотнесения (мэппинга) остатков. Например, в 1С материалы ведутся только на уровне партии и подразделения, при этом

само подразделение может физически представлять собой огромное складское помещение. В данном случае одна строчка остатков из 1С может превратиться в тысячи позиций для загрузки остатков SAP ERP.

- В случае наличия складской системы учёта необходимо провести соотнесение оргструктуры с SAP. Может возникнуть вопрос, как несколько зон или мест хранения будут объединены в одну зону или место хранения в SAP ERP?
- Некорректный мэппинг данных в системах. В случае, если в системах ведётся разная кодификация продукции, то при загрузке остатков возникнет ошибка. Как было описано ранее, важно определить мастер-систему.
- И, наконец, конвертация единиц измерения из исторических систем в SAP ERP.

### Перезтикетирование

Использование новых этикеток (рис. 2) для товаров и материалов на складе при переходе к SAP может превратиться в большую проблему. Изначально требуется оценить предполагаемый объём материалов, для которого нужно выполнить повторную наклейку этикеток (перезтикетирование). Важно чётко понимать новую организационную структуру и учёт материалов в SAP WM. Кроме того необходимо определиться, на каком уровне материалов будет клеиться этикетка. Например, одна этикетка будет наклеиваться на всю партию материала или, если внедрена функциональность HU, этикетки могут клеиться на паллеты/коробки/материалы. Принимая во внимание учёт единиц обработки, размеры склада и складские остатки, число этикеток может измеряться десятками тысяч.

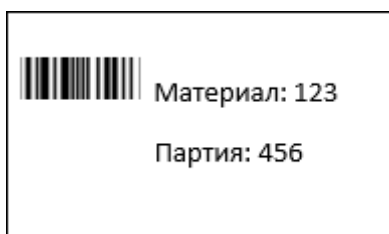


Рис. 2. Пример этикетки

Все это может стать непосильной задачей для складских сотрудников в случае, если стоит задача обклеить все материалы до промышленного старта SAP-системы. Ситуация может усугубляться в связи с тем, что на складе есть высокоуровневые стеллажи, поэтому для того, чтобы добраться до верхних уровней будут необходимы высотные фронтальные погрузчики. Даже если на обклейку этикеток выделить все

силы предприятия, но на складе присутствует только один фронтальный погрузчик, скорость переклеивания будет весьма невысокой [1].

Не стоит забывать, что обклеенный материал потребуется положить обратно на первоначальное место хранения для целей SAP. Поэтому высока вероятность допущения ошибки, так как паллеты обычно этикетируются не рядом со стеллажом, а переносятся в специальную зону переупаковки. Если в процессе переупаковки участвует несколько человек и все они делают переупаковку по одному и тому же стеллажу, паллеты могут быть поставлены не на то место. Помочь предотвратить подобные ошибки может проведение тестовой перетикетировки части запаса.

### **Обучение пользователей**

Функциональность WM довольно сложна для понимания ключевых и конечных пользователей, особенно если идёт внедрение данного функционала в комбинации с единицами обработки. Поэтому необходимо обратить особое внимание на обучение ключевых пользователей, а функциональному консультанту по возможности принять участие в обучении конечных пользователей. При обучении пользователей стоит использовать ассоциативные примеры. Например, взять несколько пластиковых стаканов с водой и сказать, что каждый из них представляет собой паллету, а с точки зрения SAP – это HU [2]. Далее наклеить бумажки для напоминания – это будут этикетки, потом перелить воду из одного стакана в другой, поясняя, что это есть процесс переупаковки.

### **Карманные ПК для обработки этикеток**

Обработка этикеток подразумевает применение PDA (Personal Digital Assistant – карманный персональный компьютер) [3]. Для реализации работы PDA может быть несколько вариантов:

- использование PDA на базе системы SAP;
- применение программного обеспечения сторонних производителей;
- создание собственной оболочки для работы PDA.

В случае использования решения SAP нет необходимости покупать дополнительные лицензии и платить за поддержку оборудования, а также держать штат программистов для поддержки и развития локального решения. Опыт показывает, если на экране PDA располагается больше трёх кнопок или полей ввода данных, то работать с ним становится очень неудобно: сложно попасть в нужную кнопку. Кроме того, необходимо проверить дальность работы PDA: если карманные ПК не могут сканировать штрих-код на расстоянии 1 метра от объекта, то изъятие материалов с верхних полок будет проблематично.

## Обработка и печать этикеток

Для работы с этикетками потребуется закупка специальных принтеров, самоклеящейся бумаги и краски. При физической работе с этикетками можно сформулировать следующие советы:

- проверяйте PDA читаемость штрих-кода, напечатанного на этикетке;
- определив содержимое этикетки, не меняйте его;
- не выводите на этикетке информацию, меняющуюся со временем, например, количество. Это одно из первых требований ключевых пользователей, но так как количество изменяется со временем, то потребуется каждый раз перепечатывать этикетку. Поэтому лучше использовать PDA для сканирования этикетки и выяснения актуального количества.

На этапе проектирования SAP ERP по модулю WM лучше создать отдельную спецификацию для всех этикеток. При разработке спецификации возможно увеличение количества требуемых форматов этикеток. Например, каждое подразделение может захотеть иметь свою собственную этикетку или сделать этикетки для каждого бизнес-процесса, определить отдельный формат этикетки для закупаемых и производимых материалов и т.д.

При этом этикетки могут быть разных размеров и соответственно нужно будет настраивать принтеры под каждый формат отдельно. Лучше всего иметь один общий формат этикетки для всего предприятия с заранее определённой универсальной информацией на ней. В качестве информации на этикетке можно выводить следующие данные:

- код материала;
- наименование материала;
- партия материала;
- правила работы с материалом, в случае если материал опасный.

И в заключении, не забудьте подготовить этикетки для складских мест, чтобы можно было быстро помещать материалы в ячейки.

## Литература

1. Логистика / Дыбская В.В. и др. - М.: Эксмо, 2009. - 944 с.
2. Официальный курс SAP «TERP10 - SAP ERP Integration and Business Processes». <https://training.sap.com/shop/course/terp10-sap-erp-integration-of-business-processes-classroom-010-us-en/>

3. Лодон Дж., Лодон К. Управление информационными системами / Пер. с англ.; под ред. Д. Трутнева. - СПб.: Питер, 2005. - 910 с.

### **Выходные данные статьи**

Терентьев И.М. Сложности внедрения WMS-систем на примере SAP ERP  
// Корпоративные информационные системы. - 2018. - №1. - С. 40-45. - URL:  
<http://corpinfosys.ru/archive/issue-1/45-2018-1-wms>

### **Об авторе**



**Терентьев Илья Михайлович** - эксперт по системам управления складами. Сертифицированный консультант SAP по модулю MM. Принимал участие более чем в 10 проектах имплементации корпоративных систем. Имеет обширный опыт внедрения складских систем в проектах «с нуля», а также тиражирования. Регулярно публикует статьи в электронно-сетевом журнале «САПер». Электронный адрес: [ilya.terentiev89@gmail.com](mailto:ilya.terentiev89@gmail.com)