

ФОРМИРОВАНИЕ УНИВЕРСАЛЬНЫХ ТРЕБОВАНИЙ К ПОЛЬЗОВАТЕЛЬСКИМ ПРОГРАММАМ ПРИ ПОДГОТОВКЕ СПЕЦИФИКАЦИИ НА АВАР-РАЗРАБОТКУ

Степанов Дмитрий Юрьевич
stepanov@mirea.ru, mail@stepanovd.com

«Московский государственный технический университет радиотехники,
электроники и автоматики» (МГТУ МИРЭА)

Аннотация: используя обобщенную структуру описания программ, формулируются универсальные требования к пользовательским разработкам для подготовки технической спецификации.

Ключевые слова: спецификация на разработку, АВАР, SAP, КИС.

UNIVERSAL REQUIREMENTS TO ABAP-PROGRAMS WHILE PREPARING SOFTWARE MODULE SPECIFICATION

Dmitry Yu Stepanov
stepanov@mirea.ru, mail@stepanovd.com

Moscow state technical university of radio-engineering, electronics and automation
(MSTU MIREA)

Abstract: there is general program description structure considered in the paper. Using proposed structure an approach to functional specification preparation is suggested including universal requirements to client specific ABAP-programs.

Keywords: program specification, ABAP, SAP, ERP.

А когда нужно сделать разработку? Вчера. Подобная ситуация знакома как начинающему, так и опытному консультанту SAP, независимо от функционального направления. Насколько бы ни подходило стандартное решение SAP заказчику, рано или поздно возникнет потребность в незначительной доработке системы, пусть даже самой небольшой. Именно от правильности написания спецификации на разработку зависит и скорость реализации, и качество необходимой доработки.

Как правило, процесс подготовки и реализации спецификации колеблется между двумя крайностями: функциональный консультант впервые или достаточно редко готовит подобные документы, однако опыт ответственного АВАР-разработчика позволяет сгладить неточности и шероховатости

технического задания. И, наоборот, даже самая искусно написанная консультантом спецификация будет реализовываться чрезвычайно медленно и, с большой вероятностью, ошибочно по причине отсутствия опыта со стороны программиста.

На практике баланс в подготовке и реализации спецификации достигается достаточно редко. Прослеживается следующая закономерность: чем меньше времени потрачено на написание технического задания консультантом, тем больше времени затрачивается на разъяснение программисту сути разработки и отыскание алгоритмических ошибок в процессе тестирования. Сложность программы, к сожалению, увеличивает трудоемкость тестирования разработки и может привести к тому, что большая часть программы останется «черным ящиком» как для функционального консультанта, так и для конечного пользователя. Не зря говорится, что «пожар легче предупредить, чем потушить».

Несмотря ни на что, приступаем к подготовке многострадальной спецификации. Первая сложность, с которой сталкиваемся, – это реалистичная оценка трудозатрат на формирование документа спецификации. Допустим, как-то оценили, что дальше? Описываем требования заказчика и готовим тестовые данные. Все? Разработка выполнена. Тестируем « $2 + 2 = 4$ », верно. А вот сколько будет « $2 + 3$ »? Такого задания в спецификации описано не было. Тогда в лучшем случае получим равенство « $2 + 3 = 4$ ». Сложность вторая – доскональное описание алгоритмов, проверок и данных, пусть даже вполне очевидных. Исправили, доработали. Тестируем, все верно, « $2 + 3 = 5$ ». А все ли пользователи имеют полномочия на выполнение суммирования? Это третья, но не последняя сложность. Знакомо?

Таким образом, приведенный выше пример иллюстрирует необходимость формирования универсальных требований к разработкам без акцентирования внимания на характере прикладной задачи. Предлагается применение обобщенной структуры описания программ.

ЦЕЛИ И ЗАДАЧИ

Цель данной работы заключается в формировании универсальных требований к пользовательским программам на языке SAP ABAP. Для достижения поставленной цели решаются следующие задачи:

- рассмотрение обобщенной структуры описания программ;
- обзор требований к программным разработкам;
- апробация предлагаемых требований.

1. ОБОБЩЕННАЯ СТРУКТУРА ОПИСАНИЯ ПРОГРАММ

Запуская транзакцию в SAP-системе, мы имеем дело со стандартными и пользовательскими программами. Стандартные программы входят в базовый комплект поставки системы SAP и максимально интегрированы с прочими приложениями. Пользовательские программы разрабатываются под конкретные требования заказчика и часто содержат непродуманную логику взаимодействия со стандартными компонентами SAP. Выделим разновидности пользовательских разработок, следуя [1]:

- новые программы (X,Y,Z-программы);
- новые печатные формуляры (X,Y,Z-программы печати формуляров);
- новые отчеты (X,Y,Z-отчеты);
- расширения стандартных программ, формуляров и отчетов.

Все приложения имеют схожие составные части независимо от вида пользовательской разработки, исключением являются лишь расширения. В работе [2] предложена обобщенная структура описания разработок, включающая:

- экран селекционных данных;
- экран отображения выбранных данных;
- экран результатов обработки выбранных данных.

Допустим, необходимо изменить текст определенных позиций документов материалов. Селекционный экран программы служит своеобразным фильтром, позволяющим ограничить начальную выборку данных, например, такими параметрами как «Год», «Документ материала» и др. (рис. 1а).

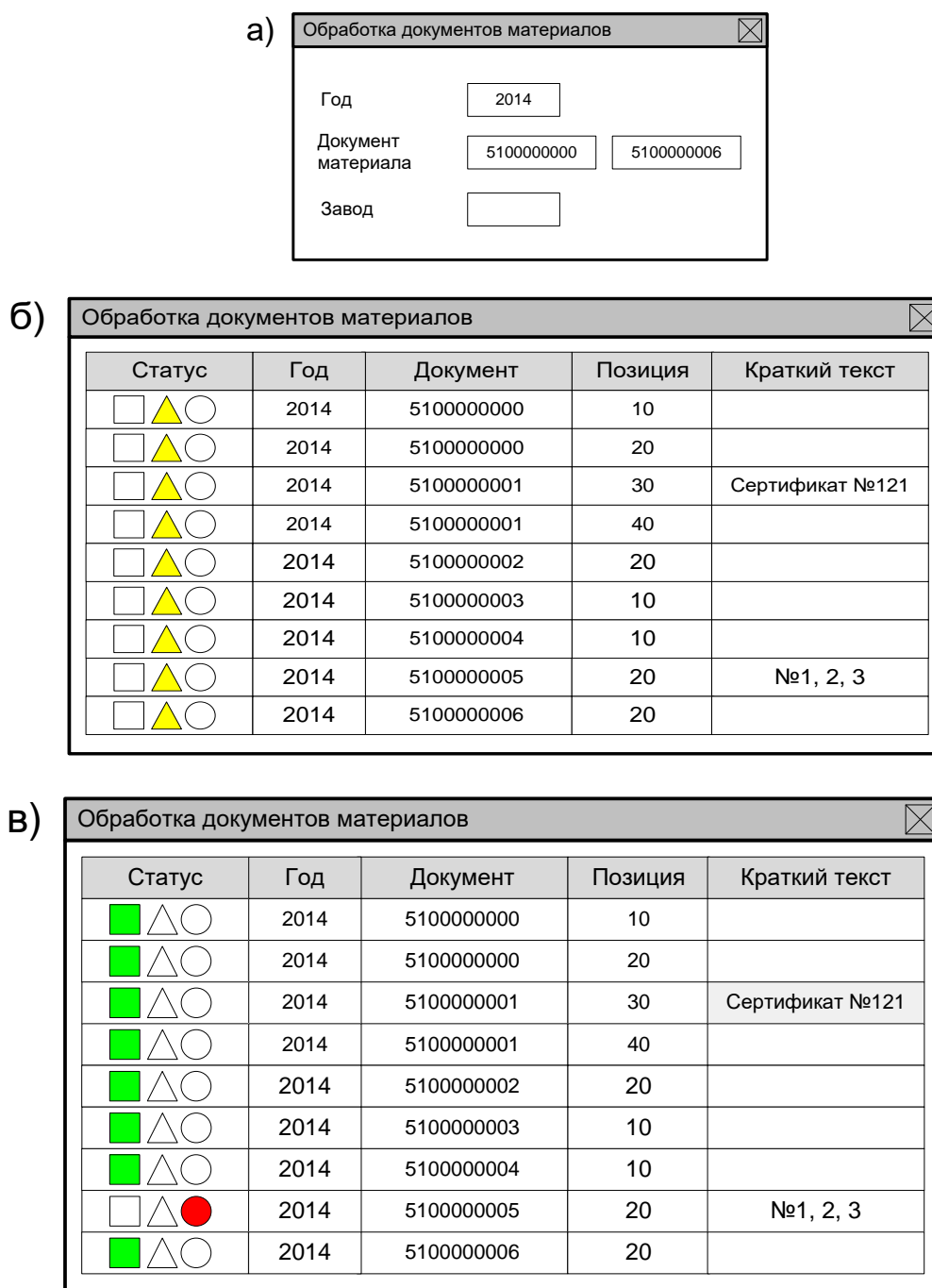


Рисунок 1. Обобщенная структура описания пользовательской программы: а) селекционный экран; б) экран отображения выбранных данных; в) экран результатов обработки выбранных данных

Выбранные данные отображаются на следующем экране программы. Что же это дает? Во-первых, проверку корректности селекции данных. Во-вторых, возможность последующей обработки выбранных данных. В-третьих, при отсутствии необходимости в отображении выбранных данных экран можно скрыть от пользователя (рис.1б).

Вручную указав необходимые позиции документов материалов, запускается процесс их изменения. Рис.1в иллюстрирует экран результатов обработки выбранных данных.

2. ОПИСАНИЕ ТРЕБОВАНИЙ К ПОЛЬЗОВАТЕЛЬСКИМ ПРОГРАММАМ В СПЕЦИФИКАЦИИ НА РАЗРАБОТКУ

Написать в спецификации, конечно же, можно очень многое, а забыть – еще больше. Для исключения последнего воспользуемся обобщенной структурой описания программ, тогда спецификация на разработку будет включать следующие разделы:

- блок общей постановки задачи;
- экран селекционных данных;
- экран отображения выбранных данных;
- экран результатов обработки выбранных данных;
- блок тестовых данных.

Рассмотрим каждый из разделов спецификации подробнее и сформулируем универсальные требования к реализуемым пользовательским программам.

Блок общей постановки задачи

Что, собственно говоря, не так, и какой подход предлагается для решения? Ответы на данные вопросы должны содержаться в разделе общей постановки задачи. Прочитав постановку задачи, человек, пусть даже не вполне знакомый с системой SAP, должен понять суть проблемы и уловить основные мотивы

подготовки документа. Например, можно воспользоваться следующей структурой описания [3]:

- исходное поведение системы (модель AS IS);
- описание проблемной области;
- целевое поведение системы (модель TO BE);
- предлагаемое решение.

Излишняя детализация решения на данном этапе преждевременна и необходима лишь для понимания предлагаемого подхода к решению. По усмотрению разработчика может быть предложен иной подход к реализации задачи. Значительно упрощает понимание как самой задачи, так и предлагаемого решения описание, представленное, например, в графическом виде (рис.1).

Подготовка данного раздела закреплена целиком и полностью за функциональным консультантом, в то время как описание последующих блоков, за исключением тестовых данных, может находиться в сфере ответственности разработчика.

Требования к селекционному экрану программы

Первый шаг – запуск программы на выполнение. Запустили транзакцию «Обработка документов материалов», а сработала печать формуляра? Неправильно, наименование транзакции должно отражать конечный результат работы программы. Что, собственно, следует из определения транзакции [4]. Возвращаясь к приведенному примеру, корректнее было бы назвать транзакцию «Печать формуляра».

Как обобщить программу, решая вполне конкретную задачу? Ответ, никаких константных переменных. Константы выносятся в дополнительную настройку [5]:

- добавив поля на селекционном экране программы;
- используя ракурс константных величин TWARV (тр. SM30);
- на основе собственной настроечной таблицы;

- при помощи пользовательских наборов (тр. GS01).

Каждый из способов имеет свои сильные и слабые стороны. Например, вывод всех возможных параметров на селекционный экран в значительной степени отягощает понимание программы, однако можно воспользоваться вариантом транзакции для скрытия части полей. Ракурс TWARV используется всеми модулями системы SAP, что чревато переносом в продуктивную систему «лишних» записей. Применение собственной настроенной таблицы исключает последнее, однако требует трудозатрат по созданию нового объекта системы. Возможность ведения лишь одного признака является ахиллесовой пятой применения пользовательских наборов.

Имея ограниченные права, возможно ли увидеть все документы? Да. Разрабатывая пользовательскую программу, необходимо позаботиться о проверке полномочий непосредственно в самом тексте программы (рис.2а), иначе максимум, что вы получите, это проверку на код запускаемой транзакции (рис.2б).

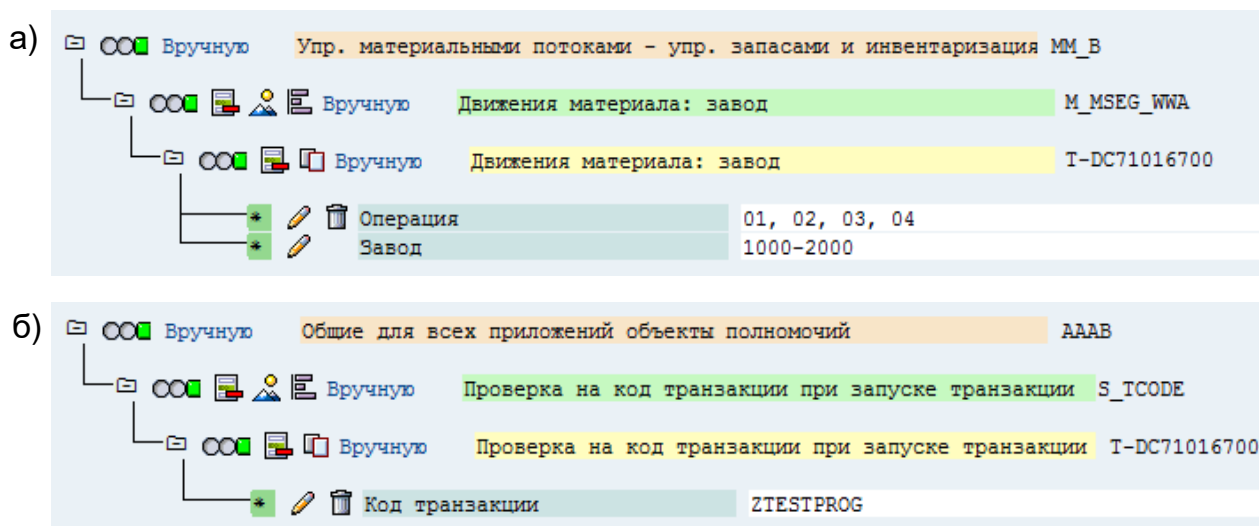


Рисунок 2. Объекты полномочий на основе:
а) прикладного компонента; б) кода транзакции

Таким образом, основными требованиями при описании селекционного экрана пользовательской программы являются: задание корректного

наименования транзакции; определение и перенос всех константных переменных программы в настройку; указание логики проверки полномочий пользователя.

Требования к экрану отображения выбранных данных

Второй шаг – получение результатов выборки данных. Почему программа работает так медленно? Чем сложнее алгоритм обработки, тем больше времени затрачивается на поиск информации в таблицах данных [6]. В спецификации на разработку следует упомянуть, для какого объема данных будет применяться программа (документы, созданные за неделю, период или год). Способ увеличения быстродействия определит сам разработчик, например: индексация таблиц, корректировка SQL-запросов и др. Понять, насколько программа подлежит оптимизации, возможно самостоятельно, используя стандартные средства SAP (рис.3).

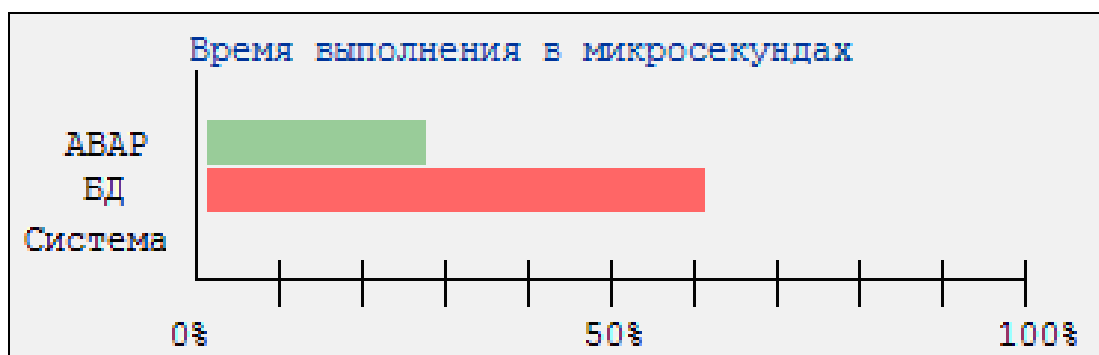


Рисунок 3. Расчет времени выполнения программы транзакцией SE30

Согласно приведенному рисунку, время поиска информации в таблицах данных эквивалентно времени обработки найденных данных. Таким образом, оптимизация алгоритма обработки позволит увеличить скорость работы программы практически вдвое.

Почему в выборку попали именно эти документы? И откуда такие цифры? Исключить подобные вопросы можно. Для этого необходимо обеспечить пользователю доступ к журналу детальной информации о селекции данных или контур обратной связи в контексте классической схемы управления [7].

Действительно, следуя данной схеме, пользователь представляется субъектом управления (СУ), программа задает объект управления (ОУ), контур отрицательной обратной связи обеспечивает СУ возможность обработки и управления ОУ для достижения поставленной цели (рис.4). В зависимости от вида пользовательской программы, предлагаются следующие способы предоставления пользователю доступа к журналу детальной информации о выборке данных:

- добавление зависимых полей в ALV-список;
- отображение шагов выбора данных на экран.

Добавление зависимых полей в ALV-список применимо при разработке пользовательских отчетов. Удобство использования ALV заключается в том, что избыточные поля можно скрыть с экрана пользователя, а при необходимости добавить вручную (рис.5а).

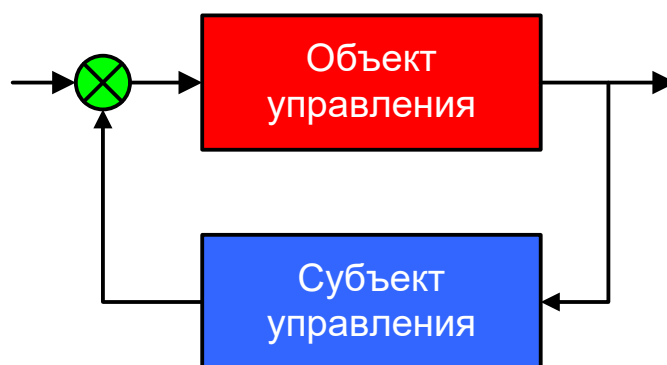


Рисунок 4. Классическая схема управления

Отображение основных этапов выборки данных важно при разработке пользовательских программ с достаточно сложными селекционными алгоритмами. В подобных случаях применим АВАР-оператор WRITE, позволяющий выводить данные на экран (рис.5б). К сожалению, в этом случае невозможно отображать и скрывать информацию так же гибко, как при использовании ALV-списка, однако допускается следующий прием. Если добавить на селекционный экран программы индикатор необходимости

отображения журнала выборки данных (рис. 1а), тогда вывод информации будет зависеть от значения данного индикатора.

Обеспечение приемлемого уровня быстродействия и предоставление пользователю доступа к журналу выборки данных являются ключевыми требованиями, задающими контур отрицательной обратной связи в классической схеме управления.

Требования к экрану результатов обработки выбранных данных

Третий шаг – получение результатов обработки данных. Вопрос быстродействия уже обсуждался в предыдущем разделе, поэтому на данном шаге ограничимся лишь фиксацией требования по обеспечению приемлемой скорости обработки данных.

Какие документы были созданы, изменены или удалены? Ответить на заданный вопрос поможет журнал обработки данных (рис.6). Журнал должен отображаться после выполнения программы и содержать список обработанных документов, который при необходимости можно скопировать в буфер обмена и далее обработать стандартными средствами SAP. Кроме списка обработанных документов, журнал может содержать информацию об исключительных ситуациях: предупреждениях и ошибках, возникших в процессе работы программы.

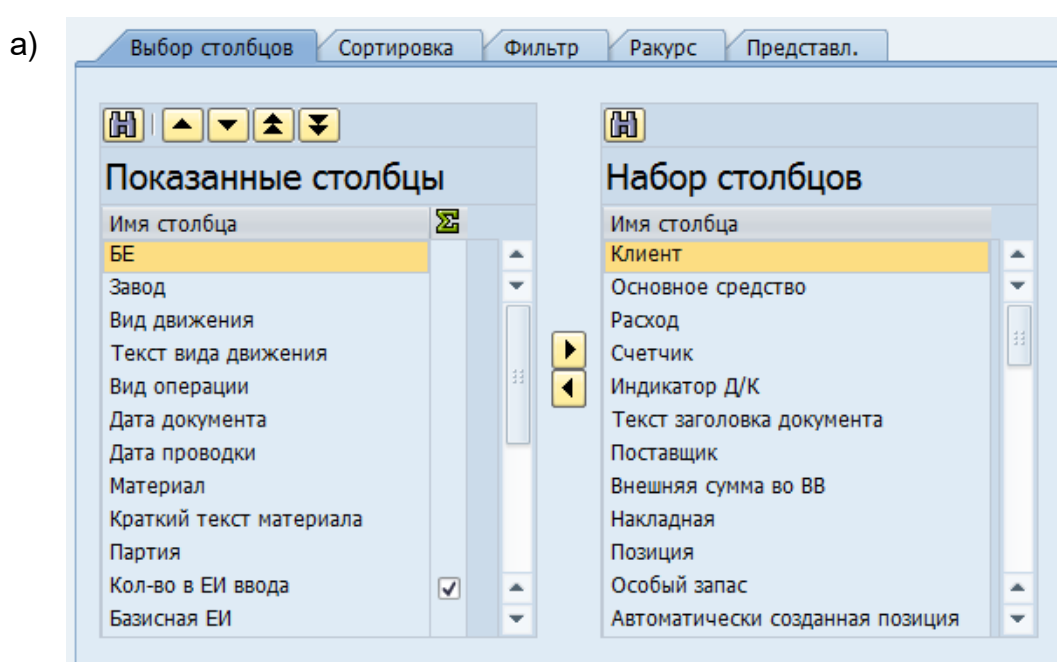
Возможно ли отыскать созданные документы? Конечно. Во-первых, список созданных документов должен отображаться в журнале обработки данных. Во-вторых, в создаваемых документах необходимо указывать идентификатор, например, в поле краткого текста: код разрабатываемой программы, дату создания и прочие технические сведения. Тогда пользователь без труда сможет найти созданные документы через стандартные SAP-отчеты, используя указанный идентификатор.

В процессе работы программы произошла ошибка, что делать с уже созданными документами? Здесь возможно несколько решений:

- прекратить работу программы и вручную удалить документы;

- автоматически удалить созданные документы и прекратить работу.

Удаление документов вручную осуществляется с использованием описанного выше способа поиска созданных документов. Автоматическое удаление созданных документов в случае возникновения ошибки необходимо закладывать в логику работы разрабатываемой программы. В любом случае журнал обработки должен содержать информацию о созданных и удаленных документах, а также возникших системных сообщениях.



б)

*** Старт программы ***

Успешно обработаны следующие документы материалов за 2014 год:

5100000000

5100000001

5100000002

5100000003

5100000004

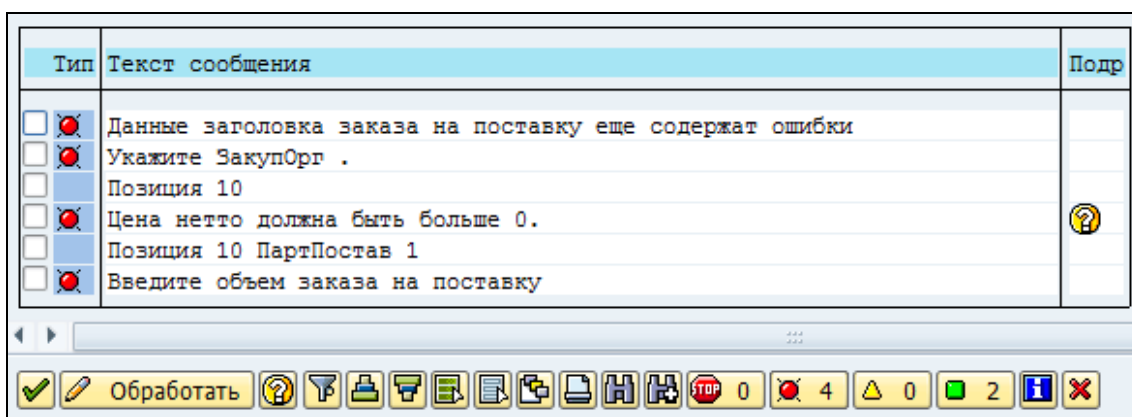
5100000005

5100000006

*** Окончание программы ***

Рисунок 5. Детальная информация о селекции данных с использованием:
а) ALV-списка выбора полей; б) оператора WRITE

Следовательно, основными требованиями при описании экрана обработки выбранных данных являются: обеспечение приемлемого быстродействия программы; вывод журнала обработки данных; указание идентификатора в создаваемых документах для возможности ручного поиска и обработка исключительных ситуаций, что определяет контур отрицательной обратной связи для управления программой пользователем.



Тип	Текст сообщения	Подр
<input type="checkbox"/>	Данные заголовка заказа на поставку еще содержат ошибки	
<input type="checkbox"/>	Укажите ЗакупОрг .	
<input type="checkbox"/>	Позиция 10	
<input type="checkbox"/>	Цена нетто должна быть больше 0.	?
<input type="checkbox"/>	Позиция 10 ПартПостав 1	
<input type="checkbox"/>	Введите объем заказа на поставку	

Toolbar: Обработать, ?, Print, Copy, Paste, Undo, Redo, Stop 0, Error 4, Warning 0, Success 2, Help, Close.

Рисунок 6. Журнал обработки документов

Блок тестовых данных

Необходимо проверить корректность работы разработанной программы, как это сделать? Блок тестовых данных предназначен для ответа на данный вопрос. Выделяют следующие этапы тестирования разработок [8]:

- функциональное тестирование;
- интеграционное тестирование.

Первый вид тестирования обеспечивает контроль правильности работы программы в рамках заданного функционального направления. Например, тестирование программы по загрузке движений материалов успешно завершено, если работа программы приводит к корректному увеличению или уменьшению складского запаса.

Интеграционное тестирование демонстрирует, насколько правильно результаты работы программы одного функционального направления

отразились в другом модуле системы. Возвращаясь к вышеописанному примеру, тестирование покажет, обеспечила ли загрузка движений материалов формирование бухгалтерских проводок по правильным счетам и аналитикам главной книги.

Тестирование программных разработок на одной-двух записях, как правило, всегда оканчивается успешно. Оценить корректность работы программы в полной мере можно, лишь используя объем тестовых данных близкий к продуктивному. Последнее позволяет проверить, насколько реализованы требования быстродействия и при необходимости внести корректировки в алгоритмы селекции и обработки данных.

3. АПРОБАЦИЯ

Сформированные требования применялись для реализации различных видов пользовательских программ (рис.7). Апробация выполнялась в нижеприведенных организациях.

Вид разработки	Пример	Селекционный экран	Экран выбранных данных	Экран обработанных данных
Программа	Распределение 16 счетов	✓	✓	✓
Отчет	Реестр входящих счетов-фактур	✓	✓	
Формуляр	Товарная накладная ТОРГ-12	✓		✓

Рисунок 7. Применимость универсальных требований для различных видов пользовательских разработок

- Международной фармацевтической компании для реализации унифицированных печатных формуляров, согласно требованиям российского законодательства.

- Крупной российской нефтяной компании при разработке программ интеграции переменных данных между SAP ERP и сторонними информационными системами [2].
- МГТУ МИРЭА для проведения практических занятий по дисциплине «Информационные технологии», а также для разработки программного комплекса распознавания образов в квалификационной работе [9].

ВЫВОДЫ

Современные корпоративные информационные системы развиваются стремительным образом в первую очередь за счет интеграции разнородных информационных систем и технологий в единый ландшафт [10].

Таблица 1. Требования к пользовательским разработкам

Вид разработки	Требования		
	Селекционный экран	Экран выбранных данных	Экран обработки данных
Формуляр	<ul style="list-style-type: none"> ▪ Корректное наименование транзакций ▪ Общий вид решения задачи ▪ Проверка полномочий 		<ul style="list-style-type: none"> ▪ Приемлемое быстродействие
Программа		<ul style="list-style-type: none"> ▪ Приемлемое быстродействие ▪ Вывод журнала выборки данных 	<ul style="list-style-type: none"> ▪ Приемлемое быстродействие ▪ Вывод журнала обработки данных ▪ Установка метки для созданных документов ▪ Удаление созданных документов в случае ошибки
Отчет			

Внедрение подобных систем на предприятиях заказчиков требует доработки стандартного функционала. Сформулированные универсальные требования ориентированы на решение упомянутых задач (табл. 1).

ГЛОССАРИЙ

ABAP	- Advance Business Application Programming, язык программирования
ALV	- ABAP List View, средство отображения отчетов в виде списка
ERP	- Enterprise Resource Planning, управление ресурсами предприятия
SAP	- Торговая марка компании SAP AG
SQL	- Structured Query Language, унифицированный язык запросов
WRITE	- ABAP-оператор для вывода информации на экран
X,Y,Z-разработки	- Программы, не входящие в стандартный пакет SAP ERP
Модель AS IS	- Описание процесса «как есть»
Модель TO BE	- Описание процесса «как должно быть»
Транзакция, тр.	- Технический код запускаемой программы SAP

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кречмер Р., Вейс В. Разработка приложений SAP R/3 на языке ABAP/4. - М.: Лори. 1998.
2. Степанов Д.Ю. Способы интеграции данных корпоративных информационных систем. // Естественные и технические науки. - 2014. - т.69, №1.
3. Лодон Дж., Лодон К. Управление информационными системами. / Пер. с англ. под ред. Трутнева Д.Р. - СПб.: Питер. 2005.
4. Грабер М. SQL. / Пер. с англ. под ред. Вендров А. - М.: Лори. 2003.
5. Точенюк О. Практические рекомендации по оптимизации программ на ABAP. // Сапёр. - 2010. URL: www.sapland.ru/articles/stats/2010/1/Prakticheskie_rekomendatsii_po_optimizatsii_programm_na_ABAP.html
6. Закорюкин В.Б. Надежность, эргономика и качество АСОИУ. - М.: МИРЭА. 2006.
7. Егоров А. Основы теории управления. - М.: Физматлит. 2007.
8. Кнут Д. Искусство программирования, том 1. Основные алгоритмы. / Пер. с англ. под ред. Козаченко Ю. - М.: Вильямс, 2006.

9. Степанов Д.Ю. Методы и алгоритмы распознавания образов с использованием древовидных представлений многоканальных изображений: Автореф. дис. кан. тех. наук / МГТУ МИРЭА. - М.: 2013.
10. Хетагуров А.Я. Проектирование автоматизированных систем обработки информации и управления. - М.: Высшая школа, 2006.

СВЕДЕНИЯ ОБ АВТОРЕ

к.т.н. СТЕПАНОВ ДМИТРИЙ ЮРЬЕВИЧ



Родился в 1984 г. Окончил факультет кибернетики Московского Государственного Технического Университета Радиотехники, Электроники и Автоматики (МГТУ МИРЭА) в 2008 г. В период 2008 – 2011 гг. обучался в аспирантуре МГТУ МИРЭА. Защитил кандидатскую диссертацию по тематике распознавания образов. Лауреат премии имени д.т.н. В.С. Корсакова-Богаткова. Сертифицированный консультант SAP по направлениям Business Foundation & Integration, SCM – Procurement и SRM – Procurement & CCM. Принимал участие в реализации проектов по функциональности MM, SD, DMS, CRM и SRM. В настоящее время преподает в МГТУ МИРЭА. Область научных интересов: системный анализ и обработка информации.