

УДК 004.01
DOI: 10.15827/0236-235X.131.385-395

Дата подачи статьи: 21.02.20
2020. Т. 33. № 3. С. 385–395

Особенности проектирования и программирования при создании информационных систем

*Р.Д. Гутгарц*¹, д.э.н., профессор, gutgarc@gmail.com

¹ *Иркутский национальный исследовательский технический университет (ИРНИТУ), г. Иркутск, 664074, Россия*

На рынке функционального ПО, предназначенного для автоматизации различных задач и уровней управления на предприятиях и в организациях, представлено большое количество готовых проектных решений. Такие приложения, как правило, позиционируются как типовые и обеспечивают адаптацию к конкретным условиям на объекте автоматизации путем настройки специальных параметров и (или) сравнительно небольшого объема дополнительного программирования.

Однако количество хозяйствующих субъектов велико, все они отличаются производственными мощностями, ассортиментом продукции и другими характеристиками. В рыночной экономике каждый из них заинтересован в сохранении и развитии своих конкурентных преимуществ, поэтому не всегда возможно найти подходящую автоматизированную информационную систему, которая удовлетворяла бы всем требованиям. И тогда возникает необходимость создать систему в соответствии с индивидуальным заказом. В таких случаях ее качество будет полностью зависеть от правильности сформулированных требований со стороны заказчика и профессионализма разработчика.

Требования сводятся к алгоритмам для решения задач, а сами алгоритмы определяются и согласуются на этапе проектирования информационной системы. При программировании выполняется только кодирование алгоритмов, то есть их запись в виде формальных конструкций на алгоритмическом языке, пригодных для исполнения на компьютере.

В статье показана реальная зависимость этапа программирования от этапа проектирования, поскольку программирование всегда вторично по отношению к проектированию. Исключением является использование технологий для быстрой разработки приложений, когда между двумя этапами образуется неразрывная связь и по существу они объединяются в один.

Доказана важность проектирования и рассмотрены некоторые его особенности. Для иллюстрации приведены примеры проектных решений.

Ключевые слова: проектирование информационных систем, этапы создания информационных систем, жизненный цикл информационной системы, особенности этапа проектирования, особенности проектирования документов, программная инженерия.

Очень динамичный ритм жизни и ускорение темпов развития информационных технологий способствуют определенным изменениям в реализации проектов, связанных с решением задач автоматизации.

Практика показывает, что сегодня наблюдается нивелирование роли проектирования в процессе создания *информационных систем* (ИС). Тем не менее, целесообразно четко разделять процесс проектирования и процесс разработки, то есть программирования.

Современный уровень владения навыками работы на компьютере породил среди пользователей, не являющихся специалистами в области *информационных технологий* (ИТ), устоявшееся мнение, что все системы, которыми они постоянно пользуются в своей профессиональной деятельности, а также в быту (например, покупки в интернет-магазинах, заполнение налоговых деклараций и других документов,

управление картами и счетами в банках и т.п.), существуют сами по себе. Мало кто задумывается о технологии их создания. Сначала все системы были спроектированы и только потом реализованы программно. Именно в виде программ они стали доступными для применения.

Иногда под проектированием ИС понимается исключительно программирование. Однако при таком подходе отсутствует «предмет разговора с заказчиком», то есть возможность обосновать, обсудить и согласовать с ним, в частности, маршруты информационных потоков, формы существования информации и ее семантическое содержание для принятия решений и т.п. Эти и другие смежные вопросы призван объединить процесс проектирования.

В работе [1] обращается внимание на то, что на рынке функционального ПО для управления предприятиями представлено большое количество приложений, тем не менее наблюдается

снижение доверия, в частности, к ERP-системам. Одна из причин заключается в том, что компании, внедряющие такие системы, стремятся не зависеть от интеграторов. А это может привести к изменению вектора развития в проектировании и разработке ИС.

Проектирование может касаться как создания новых систем, так и корректировки уже существующих.

В ряде случаев проекты по созданию функционального ПО не являются завершенными по самым разным причинам, которые принципиально можно разделить на две группы.

1. Проблемы, связанные с недостатками этапа проектирования (стилистически некачественные и неполные функциональные требования со стороны заказчика, некорректная интерпретация таких требований со стороны аналитика от разработчика, невозможность представления требований в форме задания для программирования и др.).

2. Проблемы организационно-технологического характера, обусловленные плохой организацией работ в рамках команды проекта (нерациональное распределение и регулирование работ, недостаточный профессиональный и квалификационный уровень участников в команде, необоснованные бюджет и сроки, неправильные информационные и временные формы взаимодействия как с заказчиком, так и внутри команды, использование плохо освоенного ПО, применяемого при программировании, и др.).

Специфика этапов создания функциональных приложений

Классическая схема создания функционального программного продукта (одна итерация), пригодного для эксплуатации в условиях реального производственного объекта, может быть представлена в виде хронологической последовательности отдельных принципиальных этапов, которые показаны на рисунке 1.

Такая очередность отражает принципиальную взаимосвязь между начальной идеей и ее конечной интерпретацией, воплощенной в форме готового приложения, ориентированного на решение повседневных задач пользователя. Причем логическая связь между этапами будет соблюдаться независимо от регламентирующего документа, на основе которого выполняется проект по созданию ИС (ГОСТ Р 53622–2009, ГОСТ Р ИСО/МЭК 15288–2005, ISO/IEC 15288:2002). Интересно заметить, что некоторые авторы отмечают серьезный кризис методологий и снижение спроса на применение формализованных методологий разработки ИС [2].

Но, как бы ни назывались совокупности работ в соответствии с жизненным циклом программного продукта (этапы, стадии или процессы), согласно упомянутым стандартам все они будут соответствовать следующим логическим этапам.

Этап 1. Инициация проекта. На этом этапе осуществляется общение с заказчиком, в роли которого может выступать конкретная организационно-технологическая структура (предприятие, организация, компания, подразделение и т.п.). Впоследствии разработанные для нее проектные решения возможно представить в качестве типовых и распространять в исходном или адаптированном виде для всех хозяйствующих субъектов аналогичного назначения. Если проект инициирован компанией-разработчиком, то реальный заказчик отсутствует и в качестве объекта автоматизации выбирается обобщенная модель, отражающая деятельность некоторой фиктивной организации, но обязательно имеющей типовые функции (процессы), предназначенные для автоматизированного выполнения и характерные для объектов определенного вида. Главная цель на данном этапе – выявление функциональных потребностей, которые должны быть реализованы в будущей системе. Требования пользова-

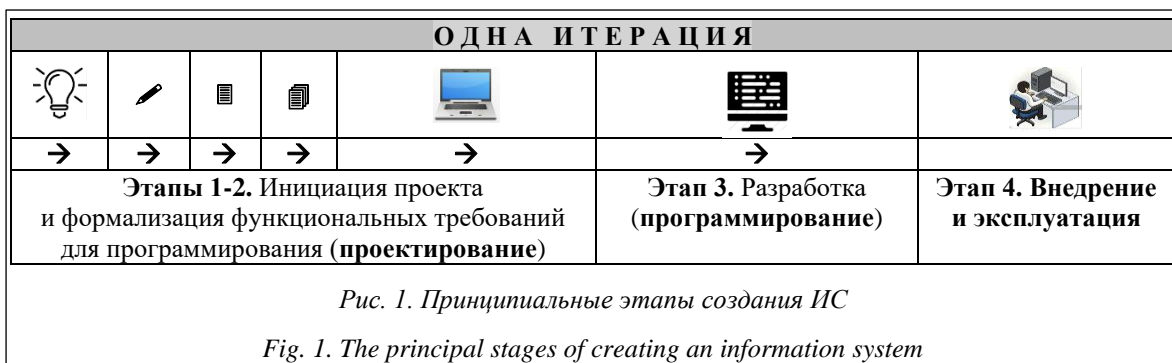


Рис. 1. Принципиальные этапы создания ИС

Fig. 1. The principal stages of creating an information system

телей должны учитывать внутрифирменные информационные обмены и информационные обмены с внешней средой.

Одна из особенностей этого этапа заключается не в простом формальном определении целей и задач системы, а в том, чтобы достижение этих целей и решение задач способствовали либо полному снятию соответствующих проблем (что возможно только за счет автоматизации), либо минимизации их последствий. Аналитическая деятельность здесь является самой важной и ответственной, так как именно ее результаты определяют стартовую позицию всего проекта.

Этап 2. Формализация требований. Изначально требования всегда сформулированы в виде «плоского» текста и программировать их невозможно, поскольку они не формализованы, то есть не описаны в форме математических и (или) логических зависимостей. Поэтому необходим этап формализации, суть которого состоит в приведении требований текстовой формы в алгоритмическую форму и (или) графические нотации, содержащие условия для программирования. Это может быть сделано при использовании любой современной инструментальной среды. Главное, чтобы результат преобразования текста в алгоритмы был понятен программистам. Поскольку единственным ресурсом, подлежащим обработке в ИС, является информация, при выявлении требований и их формализации целесообразно придерживаться следующего принципа: существуют три базовые схемы информационного взаимодействия при решении задач в рамках ИС любого функционального назначения (рис. 2).

Термин «документ» в данном контексте использован в качестве обобщающего. При решении реальных задач документом могут быть текстовый документ (с выделением реквизитов для ввода в БД или включаемых в него из БД), таблица, график (целесообразен как одна из форм для выходной аналитической информации), анимационное представление (будет полезным при отражении только определенных результатов решения) и т.п.

Независимо от применяемой схемы в качестве опций могут быть использованы вывод документа на печать, пересылка результатов по электронной почте, передача документа на архивное хранение, размещение информации в интернет-среде для публичного просмотра и другие.

Наличие предложенных схем позволяет унифицировать большинство наименований

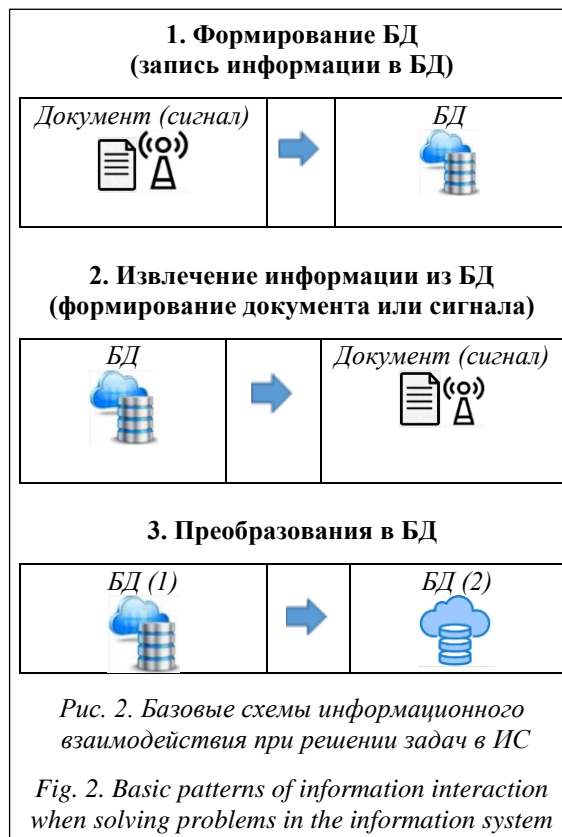
задач для автоматизированного решения через классические функции управления (планирование, учет, контроль, анализ, регулирование). Это способствует упрощению процедуры начальной идентификации задач и их последующей алгоритмизации [3–5].

На втором этапе создается проектная документация. Ее структура и содержание определяются различными факторами, в частности:

- общей функциональностью проекта и ее дифференциацией (например, по подсистемам, модулям, блокам или контурам);
- опытом и профессионально-квалификационными особенностями команды проекта;
- умением участников команды грамотно «читать» и интерпретировать графические нотации;
- традициями проектной организации;
- сроками и бюджетом проекта и др.

Таким образом, второй этап заключается в подробном описании алгоритмов в виде, пригодном для программирования.

Необходимо отметить две отличительные черты этого этапа. Во-первых, в современной индустрии создания функционального ПО принято рассматривать автоматизируемые бизнес-процессы, а не задачи. Однако такой подход не вполне корректен с формальной точки зрения,



то есть с позиции алгоритмизации, что было проанализировано в [6]. Эта же особенность отражена, например, в [7]. Во-вторых, в настоящее время очень активно используются технологии программирования RAD (rapid application development – быстрая разработка приложений), для которых проектирование и программирование представляются единым целым. Однако подобный подход нельзя считать универсальным для всех типов проектов и его применение ограничивается спецификой проектов по срокам их выполнения и объемом реализуемой функциональности.

Этап 3. Программирование. Его суть – кодирование алгоритмов, отладка и тестирование составленных программ. Этот этап сопровождается составлением эксплуатационной документации.

Этап 4. Внедрение и эксплуатация. Готовое приложение устанавливается на рабочих местах сотрудников, и осуществляется обучение пользователей работе с ним. Для периода функционирования ИС характерна функция ее сопровождения, то есть поддержание регламентированных показателей работоспособности. Фактически ИС не является неизменной за все время ее применения по разным причинам, в частности:

- необходимость в небольших программных изменениях может быть обусловлена изменениями производственных и (или) управленческих процессов;
- разработчик ИС может заменить старую версию на новую в связи с применением новых технологий обработки информации (бесплатно или за дополнительную плату в зависимости от условий договора);
- в процессе работы с системой могут проявиться ошибки, не выявленные при ее разработке; в зависимости от масштаба такие проблемы возможно либо сразу устранять путем оперативной корректировки программного кода (если есть такая возможность), либо аккумулировать (если это некатастрофично) и исправлять позднее.

В исследовании [8] приведен наглядный пример того, как не учтенная в алгоритме обработка информации привела к тупиковой ситуации при использовании системы управления персоналом. В отделе персонала потребовалось изменить фамилию сотруднице, сохранив при этом ее семейный статус. Однако система не позволила поменять фамилию без указания причины. Программисты не предусмотрели та-

кую функцию, так как она не была отражена в требованиях. Интересно, что в отделе персонала и не собирались вносить это в требования, поскольку считали, что программисту и так все понятно. В результате пострадала сотрудница, которой не выдали документ для обналичивания чека.

Приведенный факт ярко иллюстрирует один очень интересный момент. Алгоритм любой сложности всегда складывается из таких элементарных конструкций, как линейный процесс, логический процесс (выбор дальнейшего пути решения при выполнении или невыполнении некоторых условий) и циклический процесс (обычный, то есть с заранее известным числом повторений и правилами изменения соответствующего параметра, или итерационный, при котором количество повторений заранее не задано, то есть вычисления продолжаются до тех пор, пока изменяемый параметр цикла находится в определенных числовых пределах и первый выход параметра за эти пределы считается окончанием цикла). Таким образом, сложность алгоритма зависит от количества логических проверок и циклов, а также их вложенности. В постановке данной задачи для смены фамилии у женщины был предусмотрен лишь один вариант (логический процесс): женщина может изменить фамилию, только выйдя замуж. Поскольку «Фамилия» является одним из полей БД о персонале, то для изменения его значения в алгоритме должны быть учтены специальные условия. В приведенном примере было предусмотрено только одно условие – замужество, то есть для смены фамилии предлагалась единственная опция, которая параллельно требовала изменить значение другого поля – «семейное положение». Но программисты не должны знать различные особенности предметных областей, в частности, управления персоналом. Учет и корректное формулирование этих особенностей является одной из важных и сложных задач специалистов собственно предметных областей. Именно отсутствие разного рода нюансов приводит к ошибкам в функциональных требованиях и, как следствие, в будущих программах.

Кроме того, в процессе эксплуатации со стороны пользователя неизбежно возникают какие-либо замечания не критического характера. Они, как правило, суммируются и впоследствии устраняются в новой версии системы или в дополнительных программных реализациях.

Особенности этапа проектирования

Проектирование ИС представляет собой процесс последовательной нормализации проектных решений на всех стадиях жизненного цикла ИС, включая планирование, анализ требований, техническое и рабочее проектирование, внедрение и эксплуатацию [9].

В работе [10] отмечается, что проектирование ИС является наиболее ответственной стадией жизненного цикла системы.

Для ИС, основанных на использовании БД, проектирование считается самой сложной и ответственной задачей [11]. Цена ошибок на начальных стадиях проектирования особенно велика, так как они могут провоцировать необходимость большого объема переделок.

В статье [1] приводятся данные по распределению трудоемкости и времени выполнения всех этапов создания ИС согласно жизненному циклу (на примере одного предприятия). Работы проектного характера (проектирование математического аппарата, формирование перечня задач экономико-математической модели, создание архитектуры программного решения, определение набора автоматизируемых бизнес-процессов и проектирование технологических карт, проектирование средств многомерного анализа, проектирование схемы данных, проектирование интерфейса) составили 68 % от общей трудоемкости проекта и заняли 75 % времени.

Именно на этапе проектирования решаются, в частности, следующие вопросы:

- время и место появления в системе сообщений информирующего, предупреждающего, указательного или иного характера, требующих от пользователя выполнения определенных действий, с которыми он может справиться либо самостоятельно, либо обратившись к специалисту (например, к системному администратору или программисту);

- внешний вид и содержание сообщений, помогающих пользователю чувствовать себя в безопасности при работе с системой (например, обработка ошибок, возможность отмены некоторой последовательности уже выполненных операций, контроль вводимой информации, минимизация ручного ввода за счет использования возможностей выбора значений из предложенного списка); популярность интерактивных подсказок оценивается экспертами в 49,4 % [12];

- проектирование словарей, справочников, классификаторов и других упорядоченных информационных совокупностей, составляющих нормативно-справочную базу; ее создание может быть самостоятельным подпроектом;

- разработка экранных форм, предназначенных для ввода и вывода информации;

- разработка карты экранов (по принципу карты сайта);

- определение всех необходимых унифицированных решений;

- оптимизация информационных потоков с обязательным акцентированием внимания на тех, которые влияют на анализ информации для принятия решений;

- разделение аналитической (отчетной) информации на обязательную (ее получение программируется заранее) и эпизодическую (ее содержание и время, когда она может потребоваться, заранее неизвестны, поэтому для ее формирования необходимо разработать специальную систему запросов, понятную функциональному пользователю);

- описание технологических аспектов сбора, передачи, хранения, обработки и использования информации;

- составление эксплуатационной документации (например, для программного продукта «ИнфраМенеджер – комплексная автоматизация ITSM процессов» объем такой документации составляет 1 870 страниц) [13].

Наличие этапа проектирования позволяет сократить риски, возникающие в связи с характеристиками проекта, в частности:

- минимизировать, а в отдельных случаях даже полностью исключить возможность изменения функциональных требований заказчика, что создаст благоприятные условия для того, чтобы оставаться в регламентированных рамках стоимости и сроков проекта;

- более обоснованно в профессионально-квалификационном плане формировать команду для реализации проекта;

- более обоснованно определять размер проекта (в функциональных единицах), качественный и количественный состав необходимого оборудования и используемых программных средств;

- предусмотреть факторы неопределенности (целей, оценки эффективности решений из-за неучета важных аспектов, недостаточной точности предвидения хода событий, например, в результате некачественно проведенного технико-экономического анализа) [14].

В работе [10] на примере ИС по учету товара на складе ИТ-отдела в системе 1С показаны ее недостатки, которые можно считать типовыми в системах самого разного функционального назначения:

- отсутствие необходимой динамически меняющейся информации;
- наличие дублирующих документов;
- проектирование форм для ввода информации без учета их быстрой обработки;
- отсутствие логических связей между соответствующими документами.

На этапе проектирования нужно обращать внимание не только на перечисленные проблемы, но и на многие другие.

В отдельных случаях в качестве проектной документации рассматривается лишь *техническое задание* (ТЗ). Однако в настоящее время содержание такого документа трактуется совершенно произвольно и может включать в себя, например, описание общей информации о проекте и (или) спецификации задач (процессов) для программирования. Использование документа под названием «ТЗ» обусловлено, вероятно, двумя факторами:

1) термин «ТЗ» у всех на слуху и используется в среде как разработчиков ИС, так и производителей (то есть будущих пользователей ИС);

2) структура и содержание ТЗ регламентированы ГОСТ 43.602-89 (сейчас данный стандарт носит исключительно рекомендательный характер).

В специализированной учебной литературе рассмотрены все грани проектирования ИС, в частности:

- инструментальные средства проектирования (CASE-средств);
- методологические и технологические аспекты проектирования [15–17];
- международные и отечественные стандарты проектирования, в том числе основанные на жизненном цикле программного продукта, а также инструментальные средства проектирования [18–20];
- сущность работ на всех стадиях (этапах) проектирования [21, 22].

Кроме того, в указанных источниках нашли отражение такие вопросы, как

- особенности проекта по созданию ИС в ракурсе организационного, информационного, программного и технического обеспечения;
- особенности проектирования документальной составляющей, включая нормативно-справочную информацию;

- составление документации;
- проектирование бизнес-процессов, включая элементы реинжиниринга;
- проектирование БД и другие вопросы.

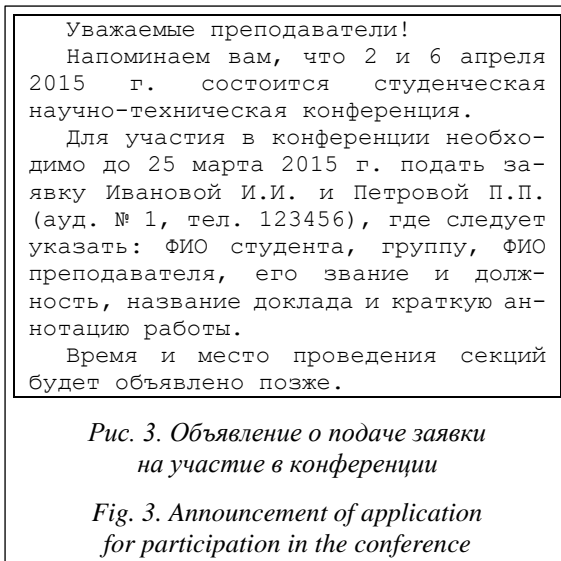
Однако в перечисленных источниках все принципиальные аспекты проектирования представлены разрозненно и недостаточно внимания уделено вопросам инженерного проектирования тех работ, которые фактически выполняются на каждом из этапов. Проектирование не показано как единый процесс, состоящий из отдельных взаимосвязанных функциональных действий. Хотя в некоторых источниках приведены примеры описания проектов, например, в [23] есть глава, посвященная разработке проектных документов. Представлены шаблоны для проведения обследования, отчет об обследовании, ТЗ, функциональная модель бизнес-решений. Постановки задач, алгоритмы, формы данных, разработка бизнес-правил приведены в форме рекомендаций. В [19] присутствует большой раздел (59 страниц), в котором очень подробно описан проект по составлению расписания учебных занятий в вузе.

Сейчас в технологиях программирования доминирует объектно-ориентированный подход в отличие от предшествующего структурного подхода, в основу которого заложен принцип алгоритмической декомпозиции. Суть объектно-ориентированного подхода заключается в том, что структура системы представляется множеством взаимосвязанных объектов, а ее функциональность терминологически выражается посредством сообщений по взаимодействию между объектами. Но как определить поведение системы, если не знать алгоритмических особенностей? Для обмена сообщениями необходимо знать логику такого «общения». А она обусловлена алгоритмами, которые обособляются и формализуются только в процессе проектирования.

Отличительная черта проектирования – его вариативность

Одной из ключевых особенностей процесса проектирования является его вариативность. Для ИС функциональная вариативность, как правило, неизменна и определяется потребностями предметной области. Напротив, вариативность возможна при проектировании внутренних компонентов, например, документов, если нет общегосударственных или отраслевых ограничений.

Рассмотрим это на простом примере. Для проведения научно-технической конференции на факультете вуза оргкомитет подготовил объявление, согласно которому все заинтересованные студенты, магистранты и аспиранты могут подать заявку на участие. В этом случае содержание объявления по существу представляло собой требования пользователей (рис. 3).



На основании собранной информации предполагалось сформировать программу конференции в традиционном текстовом виде с возможностью ее печати. Было предложено создать соответствующую структурную форму для ввода информации. Поскольку при создании ИС одним из элементов является проектирование документальных форм для ввода и вывода информации, студентам была поставлена именно такая задача.

Примечательно, что практически ни одна из представленных форм документов не повторилась. Анализ всех вариантов позволил разделить их на несколько групп:

- документы, предлагающие ввод информации в виде обычного текста вручную (рис. 4);
- документы, соответствующие вводу табличной информации в рамках Word (рис. 5);
- документы, основанные на вводе информации в БД (рис. 6);
- документы, ориентированные на ввод информации посредством web-интерфейса (форма ввода для таких документов по сути соответствовала форме на рисунке 6, хотя сама картинка, согласно специфике ее отображения, отличалась).

Проведенный эксперимент показал следующее.

Объявление с учетом возможностей последующего использования содержащейся в нем информации составлено некорректно, так как непонятна степень ее структуризации. С точки зрения сбора статистики и ее анализа, бесспорно, определенная структуризация необходима. В случае отсутствия какого-либо структурирования информация будет непригодна для дальнейшей алгоритмической обработки. Если структуризация не предполагается, каждый участник конференции будет представлять информацию в произвольной форме. Тогда организаторам придется тратить время на унификацию всех форм, поскольку программа конференции должна быть представлена в некотором упорядоченном виде.

Каждый студент, принимающий участие в эксперименте, по-своему интерпретировал

ЗАЯВКА НА УЧАСТИЕ В КОНФЕРЕНЦИИ

Фамилия студента _____ ФИО научного руководителя _____
 Имя студента _____ Звание научного руководителя _____
 Отчество студента _____ Должность научного руководителя _____
 Группа _____

Название доклада _____

Краткое описание работы

Рис. 4. Документ для ввода информации вручную
Fig. 4. Document for entering information manually

Заявка
на участие в студенческой научной конференции

ФИО руководителя _____

Звание и должность _____

№	ФИО студента	Группа	Название работы	Краткая аннотация работы

Дата заполнения «__» _____ 2015 г.
Подпись руководителя _____

Рис. 5. Документ для ввода информации в таблицу

Fig. 5. Document for entering information into a table

смысл объявления и спроектировал собственную форму, что доказывает вариативность проектных решений. Одна часть документов предполагала полностью ручное заполнение без какой-либо программной обработки, другая была ориентирована на ввод структурированной ин-

формации с учетом ее алгоритмической реализации.

Приведенный пример задания показан исключительно в иллюстративных целях, поскольку сейчас имеются приложения, позволяющие полностью автоматизировать процедуру

подачи заявок на конференцию, в том числе при использовании такой информационно-технологической среды, как Интернет.

Вариативность полезна, по крайней мере, для следующих целей.

- На начальном этапе проектирования, которое называется концептуальным (схематичным), разработчики ИС могут рассмотреть несколько вариантов с целью выбора оптимального. Анализироваться при этом могут самые разные параметры, например, количество экранов и их взаимосвязь, расположение информации на экране, прогноз длительности программирования и др.

ИНФОРМАЦИЯ О РУКОВОДИТЕЛЕ

Фамилия Звание ↓

Имя Должность ↓

Отчество Телефон

ИНФОРМАЦИЯ О СТУДЕНТАХ

Фамилия Группа ↓

Имя

Отчество

Название работы

Аннотация

Добавить студента
✓

Удалить студента
✗

Отправить заявку
@

Рис. 6. Документ для ввода информации в БД

Fig. 6. Document for entering information into the database

- Для разработчика вариативность может представлять структурные аспекты работы с информацией, использование цветового оформления, особенности организации интерфейса и т.д.

- В зависимости от характера взаимодействия с заказчиком с ним могут обсуждаться варианты отдельных проектных решений.

Заключение

Развитие ИТ периодически сопровождается какими-либо изменениями в терминологии. В последние годы стало очень активно употребляться понятие программной инженерии. Однако нельзя воспринимать данное понятие как нечто существенно отличное от другой тер-

минологии, используемой в сфере проектирования ИС. Необходимо помнить, что любая функциональная программа представляет собой конечную стадию ее создания, поэтому не может быть реализована без выполнения всех предшествующих этапов проектирования.

Программная инженерия включает в себя все аспекты создания приложений, начиная с формулирования и формализации требований, создания программной реализации и заканчивая его использованием и сопровождением [24].

Таким образом, знания об особенностях проектировании ИС необходимы студентам и магистрантам определенных направлений и профилей, а также всем разработчикам функционального ПО.

Литература

1. Виноградова Е.Ю. Актуальные вопросы проектирования и реализации корпоративных систем поддержки принятия управленческих решений на предприятии // Изв. ДВФУ: Экономика и управление. 2018. № 1. С. 102–111. DOI: 10.24866/2311-2271/2018-1/102-111.
2. Зараменских Е.П. Управление жизненным циклом информационных систем. М.: Изд-во Юрайт, 2017. 431 с.
3. Гутгарц Р.Д., Провилков Е.И. О формализации функциональных требований в проектах по созданию информационных систем // Программные продукты и системы. 2019. Т. 32. № 3. С. 349–357. DOI: 10.15827/0236-235X.127.349-357.
4. Гутгарц Р.Д., Полякова П.М. Анализ особенностей формулирования функциональных требований к автоматизированной информационной системе // Программные продукты и системы. 2019. Т. 32. № 3. С. 358–367. DOI: 10.15827/0236-235X.127.358-367.
5. Gutgarts R.D. Peculiarities of functional requirements of automated enterprise resource management system. Proc. RPTSS-2018, 2018, pp. 477–485. DOI: 10.15405/epsbs.2018.12.57.
6. Гутгарц Р.Д. Роль бизнес-процессов в формировании требований к ERP-системе // Проблемы теории и практики управления. 2015. № 1. С. 118–126.
7. Трофимов В.В., Павловская Т.А. Алгоритмизация и программирование. М.: Изд-во Юрайт, 2018. 137 с.
8. Вигерс К., Битти Дж. Разработка требований к программному обеспечению; [пер. с англ.]. М.; СПб, 2014. 736 с.
9. Панфилов А.Н., Зуев В.А., Скоба А.Н. Методы и средства проектирования информационных систем и технологий. Новочеркасск: Лик, 2016. 32 с.
10. Лученко А.В., Горбаченко И.М. Проектирование информационной системы по учету товара на складе ИТ-отдела в системе 1С // Теоретический и практический потенциал современной науки: сб. науч. стат. 2019. С. 104–108.
11. Бибииков С.В., Калиниченко С.В., Обухов А.В. Концептуально-ориентированная методика исследования предметной области при проектировании информационных систем // Изв. ТулГУ. Технич. науки. 2017. № 12-2. С. 380–391.
12. Пащенко Д.С. Как мировые тенденции в проектировании информационных систем используются в отечественной практике: результаты исследования // Инновации. 2018. № 1. С. 58–63.
13. ИнфраМенеджер. Документация. URL: <https://www.inframanager.ru/download/documents/> (дата обращения: 14.01.2020).
14. Норкин О.Р., Парфенова С.С. Конфликты при проектировании информационных систем как следствие фактора неопределенности // Изв. ЮФУ. Технич. науки. 2014. Т. 155. № 6. С. 164–167.
15. Белов В.В., Чистякова В.И. Проектирование информационных систем. М.: Академия, 2015. 352 с.
16. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем. Ростов-н/Д: Феникс, 2009. 512 с.

17. Заботина Н.Н. Проектирование информационных систем. М.: ИНФРА-М, 2013. 331 с.
18. Инюшкина О.Г. Проектирование информационных систем (на примере методов структурного системного анализа). Екатеринбург, 2014. 240 с.
19. Коваленко В.В. Проектирование информационных систем. М.: Форум, 2012. 319 с.
20. Чистов Д.В., Мельников П.П., Золотарюк А.В., Ничепорук Н.Б. Проектирование информационных систем. М.: Юрайт, 2016. 260 с.
21. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления. М.: Горячая линия–Телеком, 2011. 304 с.
22. Соловьев И.В., Майоров А.А. Проектирование информационных систем. М.: Академический Проект, 2009. 398 с.
23. Грекул В.И., Коровкина Н.Л., Левочкина Г.А. Проектирование информационных систем. М.: Юрайт, 2019. 385 с.
24. Лаврищева Е.М. Программная инженерия и технологии программирования сложных систем. М.: Юрайт, 2019, 432 с.

Software & Systems
DOI: 10.15827/0236-235X.131.385-395

Received 21.02.20
2020, vol. 33, no. 3, pp. 385–395

Design and programming features when creating information systems

R.D. Gutgarts¹, *Dr.Sc. (Economics), Professor, gutgarc@gmail.com*

¹ *Irkutsk National Research Technical University, Irkutsk, 664074, Russian Federation*

Abstract. In the market of functional software designed to automate various problems and management levels at enterprises and organizations, a large number of ready-made design solutions are presented. Such applications, as a rule, are positioned as typical ones and provide adaptation to specific conditions at the automation object by adjusting special parameters and (or) a relatively small amount of additional programming.

However, the number of economic entities is large. All of them differ in production capacities, product range, and other characteristics. In a market economy, each of them is interested in maintaining and developing their competitive advantages. Therefore, it is not always possible to find a suitable automated information system (IS) that satisfies all the necessary requirements. And then there is a need to create a system in accordance with an individual order. In such cases, its quality will completely depend on the correctness of the formulated requirements on the part of the customer and the professionalism of the developer.

The requirements are reduced to algorithms for solving problems, and the algorithms themselves are defined and agreed upon at the design stage of the information system. When programming, only coding of algorithms is performed, i.e. recording them in the form of formal constructions in an algorithmic language suitable for execution on a computer.

The paper shows the real dependence of the programming phase on the design phase since programming is always secondary to design. An exception is the technology used for rapid application development when an inextricable link is formed between the two stages and they are essentially combined into one.

The importance of design is proved and some of its features are considered. To illustrate, examples of private design solutions are provided.

Keywords: design of information systems, stages of creating information systems, the life cycle of an information system, features of the design stage, features of designing documents, software engineering.

References

1. Vinogradova E.Yu. Questions of design corporate systems for support of acceptance of administrative decisions at the enterpris. *The bulletin of the Far Eastern Federal Univ. Economics and Management*, 2018, no. 1, pp. 102–111 (in Russ.). DOI: 10.24866/2311-2271/2018-1/102-111.
2. Zaramenskikh E.P. *Lifecycle Management of Information Systems*. Moscow, 2017, 431 p. (in Russ.).
3. Gutgarts R.D., Provilkov E.I. On the formalization of functional requirements in information system projects. *Software & Systems*, 2019, vol. 32, no. 3, pp. 349–357 (in Russ.). DOI: 10.15827/0236-235X.127.349-357.

4. Gutgarts R.D., Polyakova P.M. Analysis of formulation features of functional requirements to an automated information system. *Software & Systems*, 2019, vol. 32, no. 3, pp. 358–367 (in Russ.). DOI: 10.15827/0236-235X.127.358-367.
5. Gutgarts R.D. Peculiarities of functional requirements of automated enterprise resource management system. *Proc. RPTSS-2018*, 2018, pp. 477–485. DOI: 10.15405/epsbs.2018.12.57.
6. Gutgarts R.D. The role of business processes when forming ERP system requirements. *IJ of Management Theory and Practice*, 2015, no. 1, pp. 118–126 (in Russ.).
7. Trofimov V.V., Pavlovskaya T.A. *Algorithmization and Programming*. Moscow, 2018, 137 p. (in Russ.).
8. Wiegers K., Beatty J. *Software Requirements*. Microsoft Press, 2013, 672 p. (Russ. ed.: Moscow, 2014, 736 p.).
9. Panfilov A.N., Zuev V.A., Skoba A.N. *Methods and Tools for Designing Information Systems and Technologies*. Novocherkassk, 2016, 32 p. (in Russ.).
10. Luchenko A.V., Gorbachenko I.M. Designing an information system for the IT department warehouse goods accounting in the 1C system. *Collection of Articles: Theoretical and Practical Potential of Modern Science*. Moscow, 2019, pp. 104–108 (in Russ.).
11. Bibikov C.V., Kalinichenko C.V., Obukhov A.V. A concept-oriented methodology for studying a subject domain in data systems engineering. *Izv. TulGU. Eng. Sci.*, 2017, no. 12-2, pp. 380–391 (in Russ.).
12. Pashchenko D.S. Using global trends in data systems engineering in domestic practice: Research results. *Innovations*, 2018, no. 1, pp. 58–63 (in Russ.).
13. *InfraManager. Dokumentation*. Available at: <https://www.inframanager.ru/download/documents/> (accessed January 14, 2020) (in Russ.).
14. Norkin O.R., Parfenova S.S. Conflicts in the design of information systems as a result of uncertainty factor. *Izv SFedU. Eng. Sci.*, 2014, vol. 155, no. 6, pp. 164–167 (in Russ.).
15. Belov V.V., Chistyakova V.I. *Data Systems Engineering*. Moscow, 2015, 352 p. (in Russ.).
16. Gvozdeva T.V., Ballod B.A. *Data Systems Engineering*. Rostov-on-Don, 2009, 512 p. (in Russ.).
17. Zobotina N.N. *Data Systems Engineering*. Moscow, 2013, 331 p. (in Russ.).
18. Inyushkina O.G. *Data Systems Engineering (on the Example of Structural System Analysis Methods)*. Ekaterinburg, 2014, 240 p. (in Russ.).
19. Kovalenko V.V. *Data Systems Engineering*. Moscow, 2012, 319 p. (in Russ.).
20. Chistov D.V., Melnikov P.P., Zolotaryuk A.V., Nicheporuk N.B. *Data Systems Engineering*. Moscow, 2016, 260 p. (in Russ.).
21. Rudinsky I.D. *The Design Technology for Automated Information Processing and Control Systems*. Moscow, 2011, 304 p. (in Russ.).
22. Solovov I.V., Mayorov A.A. *Data Systems Engineering*. Moscow, 2009, 398 p. (in Russ.).
23. Grekul V.I., Korovkina N.L., Levochkina G.A. *Data Systems Engineering*. Moscow, 2019, 385 p. (in Russ.).
24. Lavrishcheva E.M. *Software Engineering and Programming Technologies for Complex Systems*. Moscow, 2019, 432 p. (in Russ.).

Для цитирования

Гутгарц Р.Д. Особенности проектирования и программирования при создании информационных систем // Программные продукты и системы. 2020. Т. 33. № 3. С. 385–395. DOI: 10.15827/0236-235X.131.385-395.

For citation

Gutgarts R.D. Design and programming features when creating information systems. *Software & Systems*, 2020, vol. 33, no. 3, pp. 385–395 (in Russ.). DOI: 10.15827/0236-235X.131.385-395.