

О концепции внедрения ERP-систем

А. С. Костин

В данной статье рассматривается общая концепция внедрения стандартной ERP-системы промышленного класса. Описываются основные концепции жизненного цикла функционирования системы. Приводятся процессы и фазы внедрения ERP-системы согласно методологии Oracle. Перечислены наиболее важные стандарты внедрения информационных приложений. Описана организация канонического проектирования информационной системы в соответствии с государственными стандартами РФ.

Процесс внедрения и использования ERP-системы, как и любой другой подобной компьютерной программы, состоит из определенного набора этапов. Существует несколько различных концепций структуры этапов жизненного цикла ERP-системы. Приведем некоторые из них.

Согласно упрощенной схеме внедрения ERP-системы жизненный цикл состоит из следующих этапов:

- 1) формирование требований к системе;
- 2) проектирование;
- 3) реализация;
- 4) тестирование;
- 5) ввод в действие;
- 6) эксплуатация;
- 7) сопровождение.

Методология внедрения готовых приложений AIM (Application Implementation Method) [1], предлагаемая компанией Oracle, предусматривает более подробную классификацию процессов и фаз внедрения приложения.

Схема на рис. 1 иллюстрирует разбиение задач по процессам и фазам внедрения. По горизонтали указаны процессы, по вертикали — фазы.

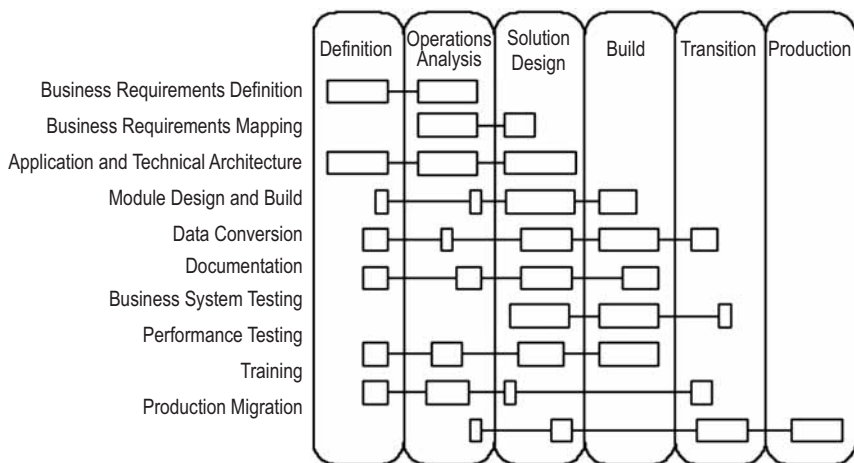


Рис. 1. Процессы и фазы внедрения готовых приложений согласно методологии Oracle

Все задачи сгруппированы в процессы по принципу общности результата. Выделяются следующие процессы внедрения:

- *Определение бизнес-требований* — результатом выполнения задач, входящих в данный процесс, является описание требований заказчика к развертываемой системе. В ходе этого процесса выявляются детальные алгоритмы, по которым происходит выполнение хозяйственной деятельности (бизнес-процессов) заказчика в области, затрагиваемой развертыванием автоматизированной системы. Затем разрабатываются детальные модели хозяйственной деятельности (бизнес-процессов) заказчика после развертывания системы, которые затем детализируются до уровня конкретных функций, выполняемых системой для каждого элементарного шага бизнес-процесса.
- *Отображение бизнес-требований* — в ходе выполнения задач, входящих в данный процесс, проводится анализ того, какая функциональность ERP-системы и каким образом может использоваться для реализации функциональных возможностей, необходимых заказчику. В этом процессе окончательно определяется, каким образом будут осуществляться хозяйственные процессы (бизнес-процессы) заказчика после развертывания системы, какая информация будет храниться в системе и какие доработки необходимо сделать, а также значения параметров настройки ERP-системы.

- *Функциональная и техническая архитектура* — в ходе этого процесса происходит построение технической архитектуры, необходимой для работы системы, а также определяются значения ключевых параметров настройки ERP-системы, касающихся архитектуры.
- *Разработка дополнительной функциональности* — в рамках этого процесса разрабатывается программное обеспечение, необходимое для реализации функциональности, отсутствующей в ERP-системе.
- *Конвертация данных* — процесс охватывает задачи, связанные с переносом данных из существующих систем (возможно в бумажном виде) во внедряемую систему. Выявляются объекты, содержащие необходимые данные, определяются методы загрузки этих данных в систему. Разрабатываются и выполняются программы переноса.
- *Документирование* — в этом процессе создается документация на систему.
- *Тестирование системы* - на основе требований к функциональности системы, собранных и детализированных в ходе процессов определения и отображения бизнес-требований, разрабатываются сценарии тестирования и производится проверка системы на реализуемость требований.
- *Тестирование на производительность* — в ходе этого процесса выполняются задачи, связанные с тестированием системы на «узкие» места по производительности.
- *Обучение* — этот процесс разбивается на две части — обучение проектной группы, с которого начинается проект по внедрению, и обучение конечных пользователей, которым проект заканчивается.
- *Ввод в эксплуатацию* — в ходе этого процесса рассматриваются все вопросы, связанные с вводом в эксплуатацию системы и ее последующим сопровождением.
- Все работы по проекту разбиваются на временные фазы. Выделяются следующие фазы:
 - *Определение* — по окончании данной фазы определяются совокупные бизнес-требования заказчика. Впоследствии они могут уточняться и видоизменяться в ходе отображения на функциональность ERP-системы, но появления новых бизнес-требований не происходит.
 - *Анализ операций* — по окончании данной фазы будущие бизнес-процессы зафиксированы и определено, как они будут реализованы с помощью ERP-системы. Так же точно определено, какие бизнес-требования не могут быть удовлетворены с помощью стандартной функциональности и какая дополнительная разработка необходима.

- *Проектирование решения* — в ходе данной фазы в частности производится создание детальных спецификаций для дополнительной разработки (функциональный и технический дизайн) и разработка сценариев тестирования.
- *Разработка* — по окончании данной фазы все дополнительные разработки завершены, приемочные тесты проведены, пользовательская документация разработана.
- *Переход к эксплуатации* — в ходе этой фазы завершается обучение конечных пользователей, производится конвертация данных и система вводится в эксплуатацию.
- *Эксплуатация системы* — это начало фазы поддержки системы. В это время выявляются и исправляются все недочеты по работе системы.

В процессе внедрения и эксплуатации любая система, в том числе и ERP, проходит различные этапы жизненного цикла. Существует три основных концепции жизненного цикла [2]:

- Каскадная модель;
- Поэтапная модель с промежуточным контролем;
- Спиральная модель.

Каскадная модель (рис. 2) предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.

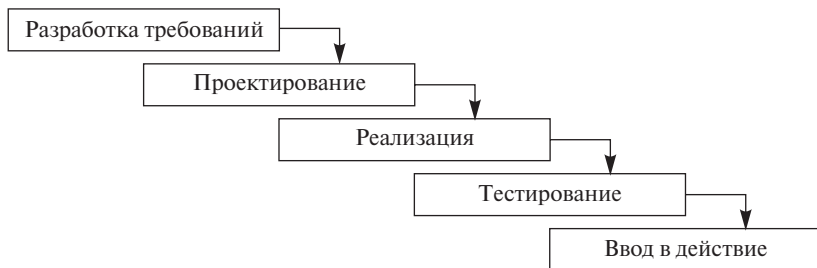


Рис. 2. Каскадная модель жизненного цикла программного обеспечения

Поэтапная модель с промежуточным контролем (рис. 3) предусматривает, что разработка информационной системы ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов

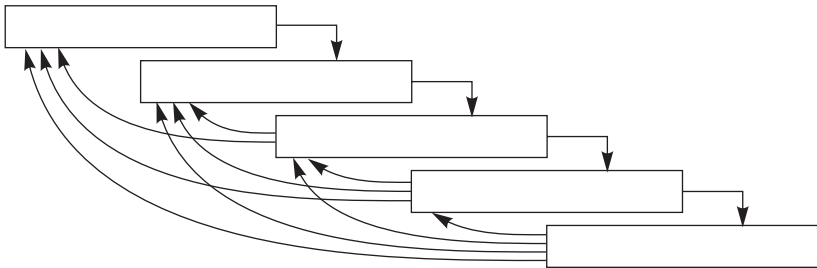


Рис. 3. Поэтапная модель с промежуточным контролем жизненного цикла программного обеспечения

разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.

Спиральная модель (рис. 4) предусматривает, что на каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка. Основное внимание уделяется начальным этапам разработки — анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).

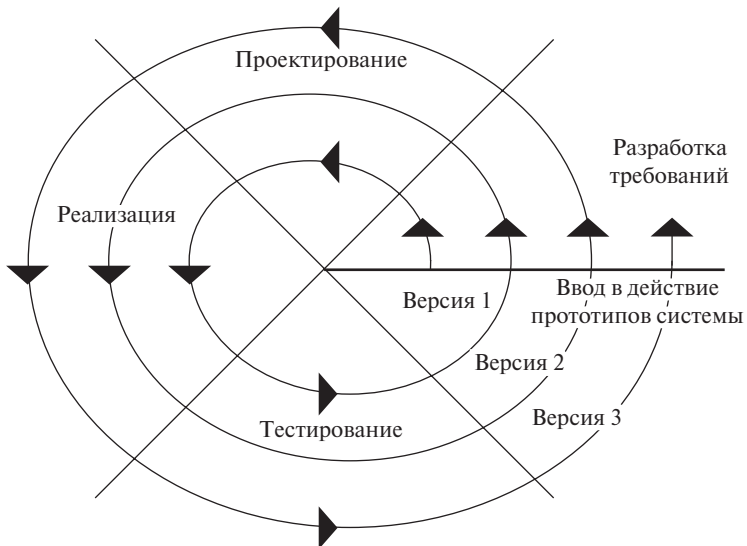


Рис. 4. Спиральная модель жизненного цикла программного обеспечения

На практике наибольшее распространение получили две основные модели жизненного цикла:

- каскадная модель (характерна для периода 1970–1985 гг.);
- спиральная модель (характерна для периода после 1986 г.).

В ранних проектах достаточно простых информационных систем каждое приложение представляло собой единый, функционально и информационно независимый блок. Для разработки такого типа приложений эффективным оказался каскадный способ. Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.

Можно выделить следующие положительные стороны применения каскадного подхода:

- на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- выполняемые в логически последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Каскадный подход хорошо зарекомендовал себя при построении относительно простых информационных систем, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования к системе. Основным недостатком этого подхода является то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания информационной системы оказывается соответствующим поэтапной модели с промежуточным контролем.

Однако и эта схема не позволяет оперативно учитывать возникающие изменения и уточнения к системе. Согласование результатов разработки с пользователем производится только в точках, планируемых после завершения каждого этапа работ, а общие требования к информационной системе зафиксированы в виде технического задания на все время ее создания. Таким образом, пользователи зачастую получают систему, не удовлетворяющую их реальным потребностям.

Спиральная модель жизненного цикла была предложена для преодоления перечисленных проблем. На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется путем создания прототипов. Каждый виток спирали соответствует созданию работоспособного фрагмента или версии

системы. Это позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали. Таким образом углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который удовлетворяет Действительным требованиям заказчика и доводится до реализации.

Итеративная разработка отражает объективно существующий спиральный цикл создания сложных систем. Она позволяет перейти на следующий этап, не дожидаясь полного завершения работ на текущем и решить главную задачу — как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

Каскадная модель предполагает разработку законченных продуктов на каждом этапе: технического задания, технического проекта, программного продукта и пользовательской документации. Разработанная документация позволяет не только определить требования к продукту следующего этапа, но и определить обязанности сторон, объем работ и сроки, при этом окончательная оценка сроков и стоимости проекта производится на начальных этапах, после завершения обследования.

Очевидно, что если требования к информационной системе меняются в ходе реализации проекта, а качество документов оказывается невысоким (требования неполны и/или противоречивы), то в действительности использование каскадной модели создает лишь иллюзию определенности и на деле увеличивает риски, уменьшая лишь ответственность участников проекта. При формальном подходе менеджер проекта реализует только те требования, которые содержатся в спецификации, опирается на документ, а не на реальные потребности бизнеса.

Существует целый ряд стандартов, регламентирующих жизненный цикл программного обеспечения, а в некоторых случаях и процессы разработки.

Среди наиболее известных стандартов можно выделить следующие:

- ГОСТ 34.601–90 — распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится описание содержания работ на каждом этапе. Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют каскадной модели жизненного цикла.
- ISO/IEC 12207:1995 — стандарт на процессы и организацию жизненного цикла. Распространяется на все виды заказного ПО. Стандарт не содержит описания фаз, стадий и этапов.

- Custom Development Method (методика Oracle) по разработке прикладных информационных систем — технологических материалов, детализированный до уровня заготовок проектных документов, рассчитанный на использование на использование в проектах с применением Oracle. Применяется CDM для классической модели жизненного цикла (предусмотрены все работы/задачи и этапы), а также для технологий «быстрой разработки» (Fast Track) или «облегченного подхода», рекомендуемых в случаях малых проектов.
- Rational Unified Process (RUP) предлагает итеративную модель разработки, включающую четыре фазы: начало, исследование, построение и внедрение. Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования. Прохождение через четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы. Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы. Суть работы в рамках RUP — это создание и сопровождение моделей на базе UML.
- Microsoft Solution Framework (MSF) сходна с RUP, так же включает в себя четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования. MSF в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений.
- Extreme Programming (XP). Экстремальное программирование (самая новая среди рассматриваемых методологий) сформировалась в 1996 году. В основе методологии командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке информационной системы, а разработка ведется с использованием последовательно дорабатываемых прототипов.

В соответствии с базовым международным стандартом ISO/IEC 12207:1995 все процессы жизненного цикла программного обеспечения делятся на три группы:

1. Основные процессы:

- приобретение;
- поставка;
- разработка;
- эксплуатация;
- сопровождение.

2. Вспомогательные процессы:

- документирование;
- управление конфигурацией;
- обеспечение качества;
- разрешение проблем;
- аудит;
- аттестация;
- совместная оценка;
- верификация.

3. Организационные процессы:

- создание инфраструктуры;
- управление;
- обучение;
- совершенствование.

Организация канонического проектирования информационной системы ориентирована на использование главным образом каскадной модели жизненного цикла информационной системы. Стадии и этапы работы описаны в стандарте ГОСТ 34.601–90 [3].

В зависимости от сложности объекта автоматизации и набора задач, требующих решения при создании конкретной информационной системы, стадии и этапы работ могут иметь различную трудоемкость. Допускается объединять последовательные этапы и даже исключать некоторые из них на любой стадии проекта. Допускается также начинать выполнение работ следующей стадии до окончания предыдущей.

Стадии и этапы создания информационной системы, выполняемые организациями-участниками, прописываются в договорах и технических заданиях на выполнение работ:

Стадия 1. Формирование требований к информационной системе

На начальной стадии проектирования выделяют следующие этапы работ:

- обследование объекта и обоснование необходимости создания информационной системы;
- формирование требований пользователей к информационной системе;
- оформление отчета о выполненной работе и тактико-технического задания на разработку.

Стадия 2. Разработка концепции информационной системы

- изучение объекта автоматизации;
- проведение необходимых научно-исследовательских работ;
- разработка вариантов концепции информационной системы, удовлетворяющих требованиям пользователей;
- оформление отчета и утверждение концепции.

Стадия 3. Техническое задание

- разработка и утверждение технического задания на создание информационной системы.

Стадия 4. Эскизный проект

- разработка предварительных проектных решений по системе и ее частям;
- разработка эскизной документации на информационную систему и ее части.

Стадия 5. Технический проект

- разработка проектных решений по системе и ее частям;
- разработка документации на информационную систему и ее части;
- разработка и оформление документации на поставку комплектующих изделий;
- разработка заданий на проектирование в смежных частях проекта.

Стадия 6. Рабочая документация

- разработка рабочей документации на информационную систему и ее части;
- разработка и адаптация программ.

Стадия 7. Ввод в действие

- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация информационной системы поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
- строительные-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;

- проведение опытной эксплуатации;
- проведение приемочных испытаний.

Стадия 8. Сопровождение информационной системы

- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

Литература

1. Методология внедрения готовых приложений AIM (Application Implementation Method). [Электронный ресурс]
// <http://www.appsoweeek.ru/modules.php?name=News&file=article&sid=19>.
2. Грекул В. И., Денищенко Г. Н., Коровкина Н. Л. Проектирование информационных систем. М.: Интернет-Ун-т Информ. Технологий, 2005, 304 с.
3. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. [Электронный ресурс]
// http://www.rugost.com/index.php?option=com_content&task=view&id=95&Itemid=53.