

БИЗНЕС И ТЕХНОЛОГИЧЕСКАЯ НЕОПРЕДЕЛЕННОСТИ В ПРОЕКТАХ ИМПЛЕМЕНТАЦИИ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ КАСКАДНОЙ И МНОГОПРОХОДНЫХ МОДЕЛЕЙ ВНЕДРЕНИЯ

к.т.н., доц. Степанов Дмитрий Юрьевич
mail@stepanovd.com, stepanov@mirea.ru

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

***Аннотация:** в статье рассматривается применимость каскадной и многопроходных моделей внедрения корпоративных информационных систем в условиях бизнес и технологической неопределенностей. Выполняется обзор каскадной, итерационной и спиралевидной моделей внедрения и разработки ERP-систем. Вводятся понятия бизнес и технологической неопределенностей, присущие проектам реализации информационных систем. Описываются базовые принципы разработки комплексных приложений в ERP-системах, включая развития и функциональности. Проводится сравнение механизмов обработки бизнес неопределенностей для уточненных требований в каскадной и Agile моделях имплементации, оперирующих запросом на изменение и формированием нового цикла разработок соответственно. Формулируется отсутствие технологической неопределенности в проектах внедрения ERP-систем и наличие высокой степени бизнес неопределенности, которую не может существенно снизить ни одна из классических моделей внедрения. Уточняется область применения водопадной и многопроходных моделей имплементации ERP-систем в проектах разработки с нуля и развития с учетом наличия бизнес-неопределенности.*

***Ключевые слова:** многопроходная модель внедрения, модель имплементации, Oracle eBS, Microsoft Dynamics NAV, методология внедрения ASAP, Oracle Unified Method, проект имплементации, eXtreme Programming, Feature Driven Development, технологическая неопределенность, кастомизации систем ERP, RICEF-программа, бизнес неопределенность.*

ВВЕДЕНИЕ

Имплементация корпоративных информационных систем, в частности, ERP-систем, ведется преимущественно с использованием преднастроенных программных решений от различных производителей. Примерами служат комплексные приложения автоматизации SAP ECC, Oracle eBS, Microsoft Dynamics NAV и 1С ERP. Использование уже апробированных программных продуктов позволяет сократить трудозатраты для их внедрения в организацию заказчика, однако вопрос кастомизации и доработки приложения остается открытым [1]. Курс на цифровизацию как частного, так и государственного сектора диктует особые требования к вопросу зрелости, а значит, технологичности и инновационности предприятий, чего не возможно представить без применения современных программных информационных систем. Темы искусственного интеллекта, машинного обучения, интернета вещей и облачных решений находят отражение в корпоративных информационных системах, что заставляет менять подход к планированию и ведению программных разработок.

Существуют несколько классических моделей разработки и внедрения информационных систем, среди наиболее популярных можно отметить прикладные методы, основанные на Agile-принципах гибкой разработки. Методология Agile позволяет иначе посмотреть на задачу программирования сложных систем, несмотря на всевозможные проектные неопределенности. Внимательно ознакомившись с литературой по

гибкой методологии [2-3], складывается ощущение, что это панацея от всего, в особенности по сравнению с классическими способами ведения разработок. Так ли это? На этот вопрос мы постараемся найти ответ в этой статье.

Цель работы состоит в демонстрации применимости каскадной и многопроходных моделей внедрения корпоративных информационных систем в условиях бизнес и технологической неопределенностей. Для реализации указанной цели будут рассмотрены следующие задачи:

- обзор каскадной, итерационной и спиралевидной моделей внедрения и разработки информационных систем;
- определение бизнес и технологических неопределенностей в проектах внедрения корпоративных информационных системах;
- исследование базовых принципов разработки сложных программ в ERP-системах;
- анализ бизнес неопределенности в каскадной и многопроходной моделях.

1. МОДЕЛИ ВНЕДРЕНИЯ ERP-СИСТЕМ

Вне зависимости от сложности и предметной направленности выделяют три базовые модели разработки программного обеспечения, предложенные более сорока лет назад и применимые к реализации ERP и ERP2-систем [4]: каскадная, итерационная, спиралевидная. Все прочие (V-модель, Scrum, Kanban и другие) являются производными от них. Важно отметить, ранее под внедрением информационной системы преимущественно подразумевалась разработка приложения «с нуля», где конфигурации отводилась минимальная роль. В настоящее время правила игры изменились: под корпоративной системой понимается кастомизированное программное решение с возможностью его доработки. Тем самым проект внедрения современных ERP/ERP2-систем заключается как в грамотной настройке, так и доработке решения. Статистика применения западных ERP-систем от компании SAP, Oracle и Microsoft на Российских предприятиях говорит о следующем [5]: 30% бизнес-требований реализуется настройками ERP-системы, 60% – требуют доработки системы, оставшиеся 10% распределяются между первыми двумя пунктами в зависимости от отраслевой специфики компании.

Каскадная модель или модель «водопад» была предложена еще в 1970 г. Реализация проекта внедрения, следуя этой модели, ведётся путём строгого выполнения задач каждого из этапов, переход к последующему этапу возможен лишь в случае успешного завершения предыдущего. Пропуск этапов, возврат к предыдущим и повторение запрещены. Проект внедрения ERP-систем на основе данной модели состоит из активностей:

- подготовка проекта;
- идентификация, анализ и приоритизация требований;
- формирование проектных решений и спецификаций на разработку;
- реализация системы;
- проведение функционально-модульного, интеграционного и приёмочного тестирований;
- миграция данных и обучение пользователей;
- переход к продуктивной эксплуатации и поддержка.

Данная модель хорошо подходит для проектов внедрения многофункциональных ERP-систем, интегрированных с множеством как внешних, так и внутренних подсистем, где даже самые незначительные обновления требований могут привести к изменениям сроков, работ и затрат. Именно поэтому водопадную модель часто применяют на проектах имплементации «с нуля» и «тиражирования», в которых объём трудозатрат достаточно велик. Основным недостатком модели служит отсутствие обратной связи между

этапами проекта. Так ошибки, допущенные в процессе концептуального проектирования ERP-решения, будут выявлены лишь на завершающей стадии приемочного тестирования, когда фактически невозможно выполнить существенные правки в коде программы. Примерами каскадной модели служат методологии внедрения ASAP (Accelerated SAP) компании SAP AG, MDSS (Microsoft Dynamics Sure Steps) от Microsoft и AIM (Application Implementation Method) как часть концепции Oracle Unified Method [6].

Итерационная модель обеспечивает контур обратной связи между этапами проекта, возможность возврата на предыдущие стадии, многократное выполнение этапов. Модель была предложена еще в 1957 г. Разработка программного обеспечения согласно данной модели сводится к разбиению этапа реализации на серию быстрых, независимых и адаптивных итераций, оперативно приносящих результаты [7]. Каждая итерация завершается демонстрацией заказчику полученного промежуточного продукта с целью скорейшего выявления замечаний и ошибок. В ходе выполнения итераций понимание содержания конечного продукта может меняться, поэтому допустимо добавление новых требований к функциональности. Продолжительность каждой итерации варьируется в пределах 1-4 недель, а число таких итераций рассчитывается на основе заранее определенных сроков завершения или бюджета проекта. Проект имплементации ERP-систем согласно предлагаемой модели состоит из шагов:

- идентификация, анализ и приоритизация изначальных требований;
- определение числа и продолжительности итераций разработки;
- реализация части ERP-системы, функциональное и интеграционное тестирование с последующей демонстрацией полученного продукта заказчику для уточнения требований, реализуемых на последующих итерациях, перенос решения в продуктивную среду (для каждой итерации);
- проведение приёмочного тестирования;
- миграция данных и обучение пользователей;
- переход к продуктивной эксплуатации и поддержка.

Из приведенного списка легко заметить, что шаги проектирования и документирования решения отсутствует, это действительно так. Однако поскольку окончание этапа продуктивного запуска завершается передачей системы службе поддержки, на практике документацию все же приходится готовить. В отличие от каскадной модели внедрения итерационная схема нашла свое применение в проектах по доработке уже реализованного решения новыми функциями, требования к которым до конца могут быть не сформулированы бизнес пользователями. К минусам спиралевидной модели относят увеличение трудозатрат, использование экспертной оценки и отсутствие четких критериев завершения проекта.

В 1986 г. была предложена спиралевидная модель внедрения. Данная модель сочетает в себе подходы как последовательной, так и итерационной моделей. Основной акцент в спиралевидной схеме сделан на обработке рисков и оценивании степени готовности решения. Спиралевидную модель нередко рассматривают как частный случай итерационной, поэтому каждый виток спирали сравним с итерацией одноименной модели внедрения и характеризуется:

- определением цели и задач;
- обработкой рисков;
- реализацией и тестированием решения;
- оцениванием результатов и планированием следующего витка разработки.

Проект внедрения ERP-систем на основе спиралевидной схемы сопоставим с предыдущей моделью с той лишь разницей, что каждый виток спирали помимо обработки рисков должен включать:

- оценку необходимости выполнения ещё одного витка спирали;
- оценку уровня понимания требований к системе;
- анализ целесообразности завершения проекта.

Обработка сроков, ресурсов, бюджета, качества, коммуникаций и прочих параметров проекта, а также оценивание необходимости выполнения следующих витков разработки требуют значительных трудозатрат. Поэтому применение данной модели целесообразно в больших проектах развития ERP-систем [8]. Для небольших подпроектов время, потраченное на обработку и оценивание параметров проекта, будет несопоставимо больше самого процесса доработки системы.

Выше отмечено, что в итерационной и спиралевидной моделях, часто называемых многопроходными, устраняется главный недостаток каскадной: отсутствие обратной связи, последнее в свою очередь не позволяет своевременно исправлять ошибки предыдущих этапов. Формализация принципов, обеспечивающих наличие обратной связи, нашла свое отражение в Agile [2]. Методология Agile включает ряд методов, к которым относят Scrum, Kanban, FDD (Feature Driven Development), а также XP (eXtreme Programming) и RAD (Rapid Application Development). Суть Agile-подхода заключается в использовании четырех базовых ценностей и двенадцати принципов, объявленных в манифесте Agile [9]. Следуя им, процесс имплементации информационных систем должен стать существенно эффективнее. К особенностям ERP-проекта на основе Agile-манифеста относят:

- максимальная вовлеченность заказчика в проектные задачи;
- многократная доработка работоспособного прототипа решения, регулярно демонстрируемого заказчику и переносимого в продуктивную среду;
- готовность и поощрение изменений в бизнес-требованиях, обеспечивающие наличие того самого контура обратной связи.

Действительно, чтобы получить комментарии к разработке периодически выполняется показ рабочей версии прототипа программы заказчику. Выявленные в ходе демонстрации недочеты включаются в следующую итерацию доработок как дополнительные бизнес-требования. Именно заказчик определяет состав требований итераций, по результатам реализации которых приоритет существующих бизнес-потребностей так же может меняться. Фактически сам клиент решает, что и когда будет реализовано с точки зрения функциональности программы.

2. НЕОПРЕДЕЛЕННОСТИ ПРИ РАЗРАБОТКЕ ПРОГРАММНЫХ СИСТЕМ

Наличие регулярной связи с заказчиком позволяет уменьшить непонимание того, каким образом должна выглядеть разработанная функция или графический пользовательский интерфейс для удовлетворения исходного бизнес-требования клиента. Назовем подобный вид неопределенности технологическим.

Определение 1. Технологическая неопределенность в проектах внедрения корпоративных информационных систем характеризует непонимание того, как разрабатываемая программа должна выглядеть и что она должна делать.

В проектах имплементации ERP-систем «с нуля» или тиражирования идентифицируют сотни, а то и тысячи требований. Приоритизация бизнес-требований преимущественно ведется по следующей схеме: законодательные требования, обязательные к исполнению на территории РФ, а также требования, обеспечивающие непрерывность бизнеса, т.е. значительно сокращающие трудозатраты конечных

пользователей, относятся к категории обязательных к реализации. Как правило обязательные бизнес-требования четко сформулированы и требуют минимум разъяснений. Требования, не относящиеся к обязательным, имеют приоритет ниже.

Именно низкоприоритетные бизнес-требования вызывают множество вопросов в ходе реализации. Слабо формализованные требования можно охарактеризовать неопределенностью, которую часто называют бизнес неопределенностью.

Определение 2. Бизнес неопределенности, присущей проектам имплементации ERP-систем, свойственны слабо формализованные, зачастую противоречивые и неполные требования, предъявляемые к бизнес-процессу и разрабатываемому приложению, автоматизирующему его.

В работе [2] технологическая и бизнес неопределенности задают неопределенность средств и конечную неопределенность. Каскадная модель внедрения предполагает сведение к минимуму оба вида неопределенности уже на начальных этапах проекта за счет введения допущений, итерационная/спиралевидная модель, напротив, допускает наличие неопределенностей и снижает их от итерации к итерации. Принимая во внимание традиционный каскадный подход и Agile к снижению неопределенности, задается применимость соответствующих моделей разработки для любого класса систем автоматизации (см. рис. 1).

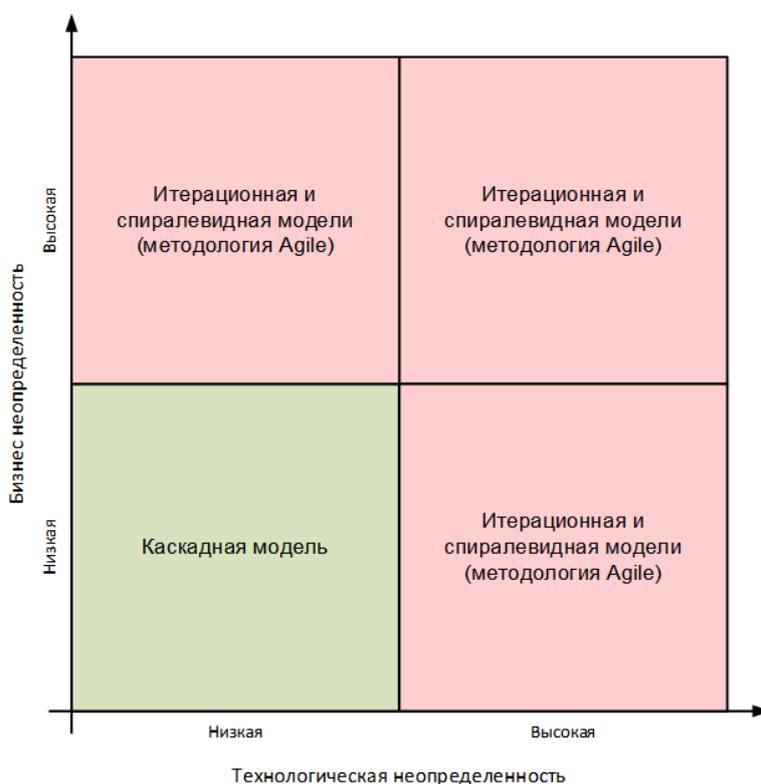


Рисунок 1. *Применимость каскадной модели, а также Agile-подходов для итерационной и спиралевидной моделей в зависимости от технологической и бизнес неопределенностей*

В прошлом разделе говорилось, что лишь 30% бизнес-требований можно покрыть за счет конфигурирования ERP-системы, остальную часть – преимущественно доработкой. Особенностью кастомизации систем класса ERP является то, что технологическая неопределенность в них сведена к минимуму. Последнее происходит из-за того, что настройка информационной системы позволяет реализовать требование преимущественно одним единственным способом без каких-либо вариаций. О чем это говорит? В первую очередь о том, что методологию Agile разумно применять в проектах доработки, но не кастомизации

ERP-системы. Во-вторых, руководствуясь данными статьи [10], в которой обосновывается целесообразность использования метода Agile Scrum в SAP-проектах, можно обобщить применимость Agile для проектов развития системы, но не ее начального развертывания.

Комплексные ERP-системы позволяют автоматизировать производственные, торговые, банковские и прочие виды организаций. Это возможно, потому что функционал информационной системы строится из заданного набора бизнес-объектов взаимодействующих друг с другом и описывающих бизнес-процессы, а также потенциальных точек расширения, используемых для доработки новых возможностей программы. Тем самым доработка любой ERP/ERP2-системы ведется в условиях жестких технических ограничений и с весьма предсказуемым объемом изменений, поэтому говорить о наличии какой-либо технологической неопределенности тоже не приходится. Суммируя определения 1-2 и гипотезы выше, сформулируем следующее утверждение.

Утверждение 1. Доработки корпоративных информационных системах, отличающихся ограниченным числом бизнес-объектов и функций для реализации новых требований, обладают минимальной технологической неопределенностью и наличием высокого уровня бизнес неопределенности.

Основной причиной наличия бизнес неопределенности является изменчивость, неоднозначность и появление новых бухгалтерских и налоговых законов, обязательных для исполнения на территории Российской Федерации. В ситуации, когда не понятно, что изменится в законе, когда изменения вступит в силу, будет ли период послабления, достаточно сложно спроектировать и реализовать программную разработку, отвечающую изначально нечетко сформулированным к ней требованиям. Последнее приводит к тому, что создаются вполне работоспособные приложения, но, как окажется в будущем, не способные полноценно покрыть непрерывный бизнес-процесс организации. Вопрос реализации приложений тесно связан с основными принципами ведения программных разработок.

3. ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ ПРОГРАММ

В работах [11-12] приводится ряд правил, которым должна удовлетворять проектируемая и разрабатываемая RICEF-программа (RICEF – сокращение возможных типов программных разработок). Например, быть масштабируемой в случае увеличения числа пользователей, гибкой к будущим изменениям, решать задачу оптимальным образом и прочее. Выделяют принципы разработки программ из следующих предметных областей:

- теория управления системами, задающая принципы программного управления, наличия контура обратной связи и обработки по возмущению;
- системный анализ, определяющий правила функциональности, развития и неопределенности;
- программирование, характеризующий модульность, функциональную избирательность, значения по умолчанию и надежность программ.

Среди указанных принципов, следует внимательно рассмотреть правила развития и функциональности. С течением времени каждая программа претерпевает изменения, т.е. развивается. Например, из-за введения новых законодательных требований, увеличения числа пользователей системы или объемов обрабатываемых данных. Принцип развития диктует необходимость проектирования ERP-программ с учетом данного обстоятельства. Именно поэтому в ERP-системах заданная программа успешно функционирует со всевозможными объектами и типами данных, а также на различных организационных уровнях.

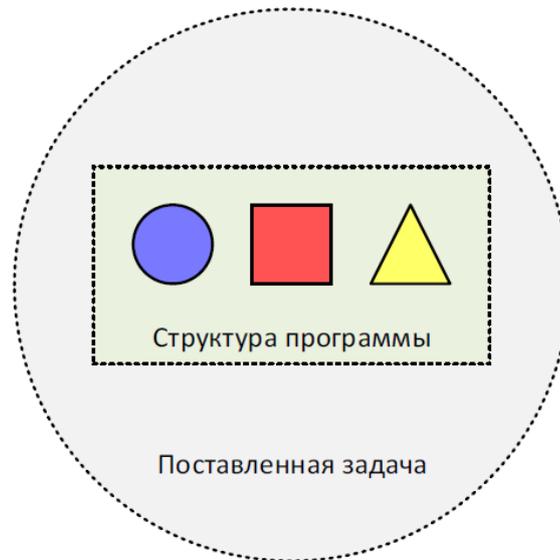


Рисунок 2. Иллюстрация принципа функциональности, где структура и содержание программы строго определены изначально бизнес-требованиями, предъявляемыми к приложению

Принцип функциональности подразумевает, что структура разрабатываемой программы корпоративной системы должна целиком и полностью определяться изначально поставленной задачей, т.е. бизнес-требованиями. Поэтому чрезмерно сложная, состоящая из большого числа кнопок, пунктов меню и подэкранов программа представляет собой результат неудачной попытки совместного решения различных бизнес-задач средствами одной программной разработки (см. рис. 2). Исключением является разработка автоматизированного рабочего места сотрудника, объединяющее множество подпрограмм, каждую из которых можно запустить независимо.

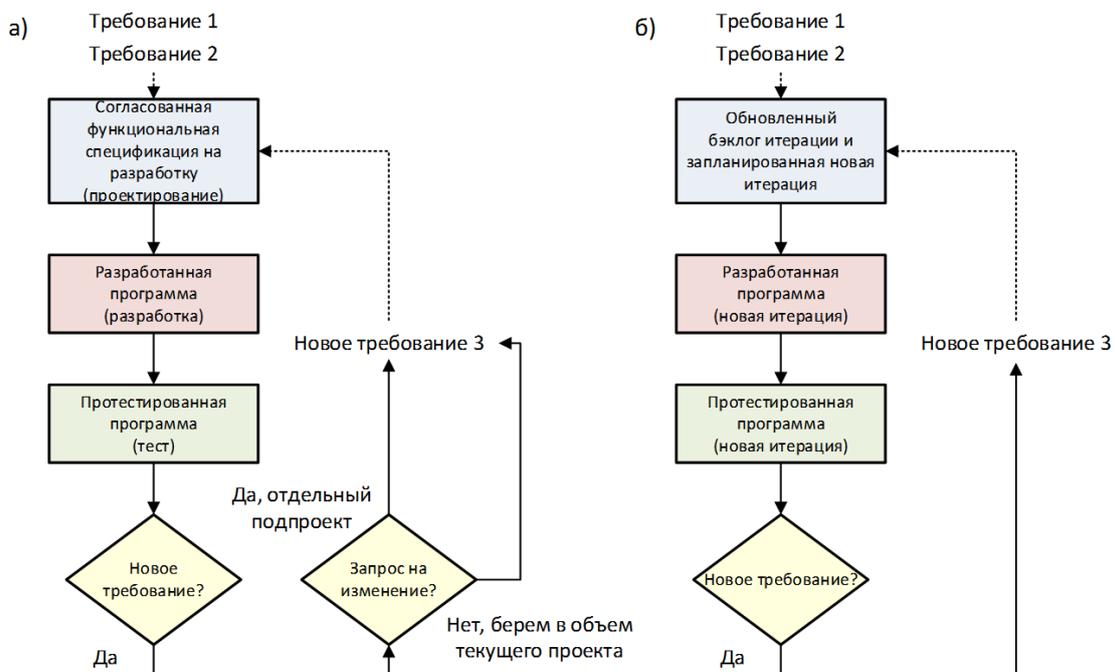


Рисунок 3. Порядок обработки уточненных бизнес-требований к ERP-разработке, следуя:
 а) каскадной;
 б) многопроходной моделям внедрения информационных систем

Используя принцип функциональности для реализации бизнес-требований в программных разработках, порядок действий на основе каскадной и многопроходных моделей внедрения можно представить следующим образом. Формируются исходные требования к приложению, для которых в последующем готовится функциональная спецификация на разработку или планируется их реализация на новой волне разработки. Далее ведется кодирование и испытание приложения, в результате чего детализируются требования, т.е. рождаются новые потребности, уточняющие исходные.

Утверждение 2. Водопадная модель в проектах внедрения ERP-систем подразумевает реализацию уточненных требований за счет отдельных независимых подпроектов путем формирования запросов на изменение, в отличие от многопроходных моделей, где новые подтребования приоритизируются и исполняются на следующем цикле разработки.

В каскадной схеме внедрения часто новые требования безвозмездно включаются в текущий объем ERP-проекта, когда суммарные трудозатраты на реализацию доработки не превосходят заданную согласованную величину, например, три человеко-дня (см. рис. 3). Следуя рисунку выше, видно, что обе модели позволяют управлять изменениями в начальных требованиях, обеспечивая при этом наглядность, гибкость и отслеживаемость хода за счет применения регламентированных процедур.

4. БИЗНЕС НЕОПРЕДЕЛЕННОСТЬ ПРИ РАЗРАБОТКЕ ERP-СИСТЕМ

Сформулированное выше Утверждение 1 ориентировано на ситуацию, когда идентифицированные бизнес-требования лишь дополняют ранее озвученные потребности. Своевременное применение принципа развития как основы для проектирования и разработки программы позволяет сохранить исходную реализованную архитектуру приложения, т.е. не нарушить правило функциональности. В реальности на проектах внедрения ERP-систем возникают бизнес неопределённости высоких порядков, например, существуют исходные требования 1-2 к процессу, для которых моделируется и кодируется программная разработка согласно принципу функциональности (см. рис 4). Бизнес неопределенность порождает кардинально новое требование 3, относящиеся к тому же процессу, но требующее значительной переделки исходного архитектурного решения. В данном случае алгоритм поведения схож с тем, что дано на рис. 3, однако возникает дилемма, как быть: значительно доработать существующую программу, но сэкономить время, нарушив правило функциональности, или разработать совершенно новое приложение, покрывающее все три требования, обеспечивающее соблюдение указанного принципа, однако требующее значительных трудозатрат?

Вне зависимости от выбранного способа как минимум возникнет необходимость в полном ретесте функционала, относящегося к исходным 1-2 требованиям. Таким образом, бизнес неопределенность приводит к значительным доработкам уже готового программного решения, обнуляя при этом предыдущие результаты. Более того, подобные изменения невозможно запланировать, а применение современных Agile-подходов не позволяет улучшить ситуацию.

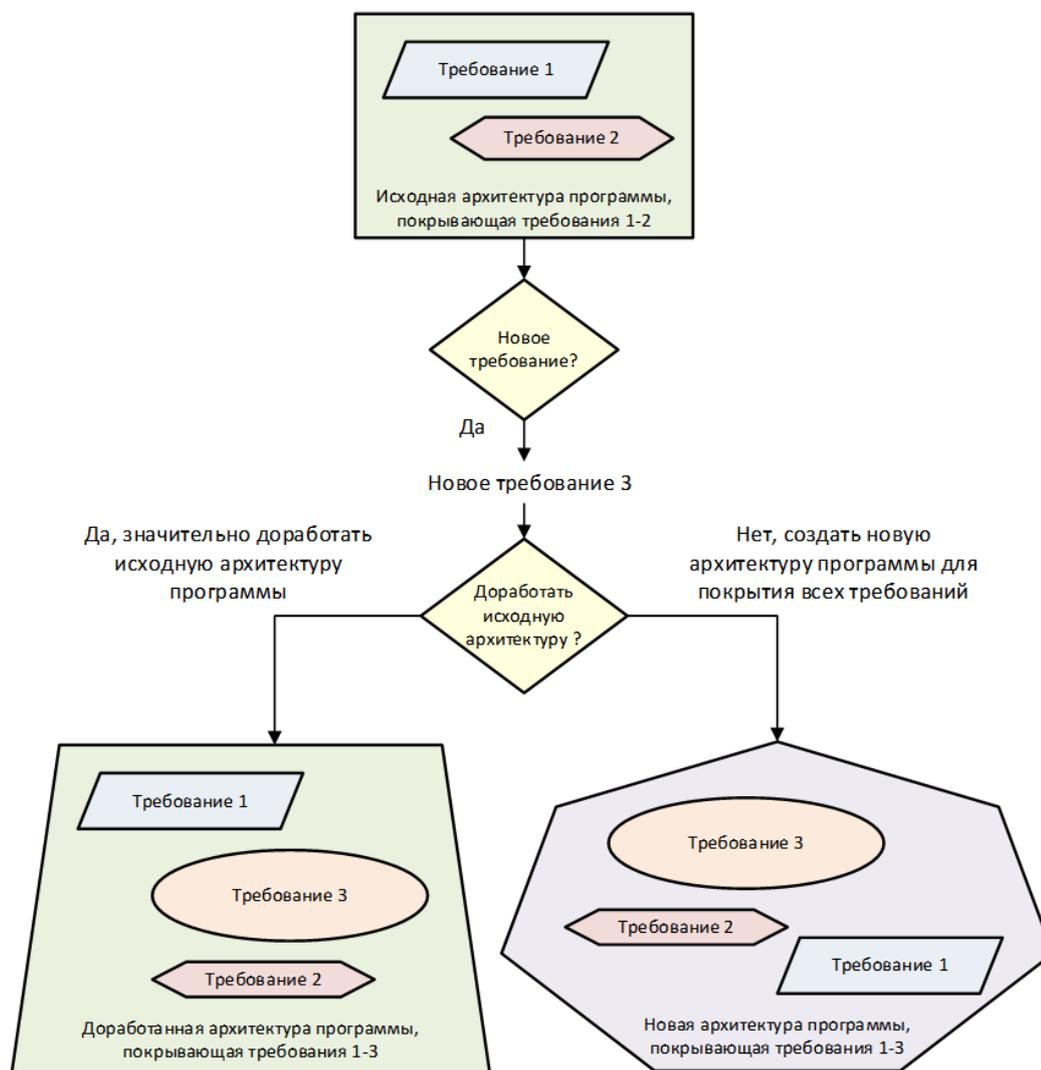


Рисунок 4. Появление нового требования при разработке ERP-системы, приводящее к изменению исходной архитектуры программы или ее полному переделыванию

Утверждение 3. Бизнес неопределенность может породить новые требования, для реализации которых необходимы значительные изменения в уже разработанных приложениях, что невозможно исключить или запланировать ни в одной из моделей внедрения ERP-систем.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Функционал современных пакетных корпоративных информационных систем ограничен заданным набором технических средств, позволяющим реализовать различные требования к бизнес-процессам. Технологическая неопределенность в проектах внедрения ERP-систем минимальна. Однако проектам присуща бизнес неопределенность, заключающаяся в размытости исходных требований, предъявляемых к разрабатываемой системе. Каскадная модель внедрения корпоративных систем применима в случае минимальной бизнес неопределенности. Средний уровень неопределенности допускает использование как на водопадной, так и многопроходных моделей, каждая из которых позволяет управлять уточненными требованиями. Высокая бизнес неопределенность нарушает основополагающий принцип функциональности при доработке программ, поэтому выбор модели внедрения не столь принципиален.

Не смотря на популярность многопроходных моделей разработки программ, в частности, Agile-подходов, существенных преимуществ по сравнению с каскадной моделью в ERP-проектах они не дают. Методология Agile ориентирована на высокий уровень технологической неопределенности, что по определению исключено в проектах имплементации корпоративных информационных систем. Более того, ни одна из классических моделей внедрения не позволяет снизить высокую бизнес неопределенность, таким образом их использование равновероятно. Водопадная и многопроходные модели позволяют одинаково качественно решать схожие задачи по разработке сложных программных продуктов, используя при этом различные организационные процедуры и методы. Принимая во внимание Утверждения 1-3, уточним применимость базовых моделей внедрения ERP-систем, изложенных в работе [10].

Таблица 1. Применимость моделей внедрения ERP-систем при минимальной технологической неопределенности

Уровень бизнес неопределенности	Модель	Вид проекта	Способ реализации
Высокий	Каскадная	Внедрение «с нуля» или тиражирование	Каскадная
		Развитие системы	Настройка
	Итерационная и спиралевидная (методология Agile)	Развитие системы	Разработка
Низкий	Каскадная	Внедрение «с нуля» или тиражирование	Настройка и разработка
		Развитие системы	Настройка
		Развитие системы	Разработка

Подводя итог проделанной работы, необходимо отметить следующее: каждая из моделей внедрения ERP-систем имеет свои преимущества и недостатки, позволяет разрабатывать программные решения, используя при этом различные способы, и, к сожалению, бессильна перед неопределенностью в бизнес-требованиях [13]. Поэтому выбор модели имплементации должен вестись осознанно: на основе регламентов и задач проекта, опыта проектной команды и технической целесообразности, а не руководствуясь модными веяниями и новинками.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Судаков В.А. Корпоративные информационные системы. М.: МАИ, 2016. 96 с.
2. Кон М. Agile: оценка и планирование проектов. М.: Альпина Паблишер, 2019. 418 с.
3. Сазерленд Д. Scrum. Революционный метод управления проектами. М.: Манн, Иванов и Фербер, 2019. 272 с.
4. Остроух А.В., Суркова Н.Е. Проектирование информационных систем. М.: Лань, 2019. 164 с.
5. Калянов Г.Н. Консалтинг: от бизнес-стратегии к корпоративной информационно-управляющей системе. М.: Горячая линия – Телеком, 2014. 210 с.

6. Samara T. ERP and information systems. Integration or disintegration. John Wiley & Sons Limited. 2018. 145 p.
7. Habadi A. An introduction to ERP-systems: architecture, implementation and impacts. International Journal of Com-puter Applications. 2017. Vol. 167. Issue 9. P. 1-4.
8. Ali M., Miller L. ERP system implementation in large enterprises – a systematic literature review. Journal of En-terprise Information Management. 2017. Vol. 30. Issue 1. P. 666-692.
9. Основопологающие принципы Agile-манифеста. URL: <http://agilemanifesto.org/iso/ru/principles.html>. Дата обращения 17.03.2021.
10. Степанов Д.Ю., Вельсовский А.В. Применение Agile Scrum в проектах SAP // Корпоративные информационные системы. 2018. № 1. С. 9-17.
11. Степанов Д.Ю. Подготовка функциональных спецификаций для разработки корпоративных информа-ционных систем на примере SAP ERP (часть 1) // Корпоративные информационные системы. 2019. №3(7). С. 29-52.
12. Степанов Д.Ю. Формирование универсальных требований к пользовательским программам при подго-товке спецификации на АВАР-разработку // Актуальные проблемы современной науки. 2014. Т. 78. № 4. С. 258-268.
13. Badewi A., Shehab E., Zeng J., Mohamad M. ERP benefits capability framework: orchestration theory perspective. Business Process Management Journal. 2018. Vol. 24. Issue 1. P. 266-294. Гаврилов Д.А. Управление производством на базе стандарта MRP2. – СПб.: Питер, 2008. – 416 с.

ВЫХОДНЫЕ ДАННЫЕ СТАТЬИ

Степанов Д.Ю. Бизнес и технологическая неопределенности в проектах имплементации корпоративных информационных систем на основе каскадной и многопроходных моделей внедрения // Высокопроизводительные вычислительные системы и технологии. – 2021. – № 2 (66). – с.118-128. – URL: <https://stepanovd.com/science/article/110-2021-2-uncertainties>.