

# ТЕХНОЛОГИЯ КОРПОРАТИВНОГО ВНЕДРЕНИЯ: АКСИОМЫ, МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА И ОБЛАСТЬ ЭФФЕКТИВНОСТИ

**Константин Зимин**, главный редактор Information Management

Кроме методологии внедрения «1С: Технология быстрого результата» (ТБР), компания «1С» разработала методологию – «1С:Технология корпоративного внедрения» (ТКВ). В отличие от ТБР она ориентирована, в основном, на крупные и масштабные проекты по созданию ИТ-систем на базе продуктов семейства «1С:Предприятие», в которых требуется существенная доработка типовой системы и изменение ее архитектуры, а также – в случае сложных структур коммуникаций и процедур принятия решений в компании-заказчике.

## Аксиомы

В основе технологии корпоративного внедрения (ТКВ) лежат четыре аксиомы — положения, принимаемые в рамках технологии истинными без

их доказательства. Они имеют смысл неких «норм менеджмента», в пределах которых происходит дальнейший разговор о подходах к управлению проектами. Дело в том, что при раз-

работке методологии авторы ориентировались на конкретную целевую аудиторию — ИТ-руководителей и менеджеров ИТ-проектов, которые в силу исторических причин не всегда хорошо понимают, в чем собственно состоит управление проектами вообще и ИТ-проектами в частности. Эти аксиомы следующие:

1. Управление проектами вообще, управление проектами в области информационных технологий (ИТ) и управление проектами по созданию ИТ-систем на продуктах семейства «1С:Предприятие» имеет значительно больше общих черт, чем различий. Вследствие этого рекомендации и требования «Руководства РМВоК<sup>1</sup>», а также различных других стандартов, имеющих отношение к управлению проектами, применимы и весьма полезны для создания ИТ-систем на продуктах семейства «1С:Предприятие».
2. Работа руководителя проекта заключается именно в управлении проектом. Вроде бы очевидное требование, но вот парадокс: часто руководитель проекта занимается чем угодно, кроме управления проектом. Причем он совершенно уверен, что его деятельность — и есть управление проектом. Надо заметить, что от руководителей проектов часто требуются знания в двух различных областях: проектного менеджмента и продукта проекта<sup>2</sup>.
3. Команда проекта, несомненно, должна иметь навыки и опыт в области создания продукта проекта. По поводу того, должен ли иметь такие навыки руководитель проекта, идут дискуссии (есть доводы и в пользу того, что руководитель проекта может не иметь знаний в предметной области проекта, и в пользу обратного), и здесь не стоит в них вдаваться. Несомненно одно: для того чтобы управлять проектом создания информационной системы (ИС), руководителю проекта необходимо разбираться в принципах и подходах к управлению проектами.
4. Профиль риска проекта в области информационных технологий является «обратным» по отношению к «обычным» проектам (рис. 1). По ходу проекта риски растут, а не падают, как это случается в «обычном» проекте (с большой материальной составляющей, на-

---

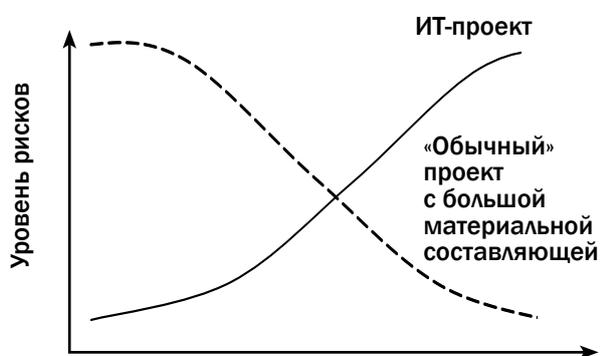
1 — Свод знаний по управлению проектами РМВоК (Project Management Body of Knowledge)

2 — Продукт проекта здесь понимается в максимально широком смысле как любой результат усилий проектной команды, имеющий как материальный характер, например, центр обработки данных, так и практически нематериальный, например, информационная система управления предприятием.

пример, постройка здания). Это происходит вследствие двух причин:

- высокой степени нематериальности выходной продукции ИТ-проекта;
- высокой неопределенности результата и неизбежно большим количеством изменений по ходу проекта.

«Обратный» профиль рисков ИТ-проекта — одно из отличий его от «обычного» проекта. Он почти полностью обуславливает предлагаемый технологией жизненный цикл проекта. Вся технология направлена на постоянную работу с рисками в ходе выполнения работ по проекту и их планомерное снижение на как можно более ранних фазах проекта.



**Рис. 1.** «Обратный» профиль рисков ИТ-проекта

5. В проекте важна работа всей команды проекта, но технология корпоративного внедрения делает акцент на управление проектом. Работа членов команды проекта (программиста, проектировщика, аналитика и др.)

направлена, прежде всего, на создание продукта проекта и освещается в других руководствах и сводах знаний. Однако именно работа руководителя проекта по управлению проектом является ключевой, она объединяет работу всех и направляет на достижение целей наилучшим образом.

## Источники и адаптация ТКВ

Технология корпоративного внедрения основана на признанных стандартах в области управления качеством (ГОСТ ИСО 9001-2011 «Системы менеджмента качества. Требования»), управления проектами («Руководство к своду знаний по управлению проектами» [Руководство РМВоК] и ГОСТ Р 54869-2011 «Проектный менеджмент. Требования к управлению проектом»), а также на опыте компании «1С» и партнеров по созданию ИС на базе продуктов семейства «1С:Предприятие» (рис. 2). В значительной степени ТКВ базируется на руководстве РМВоК<sup>1</sup>, но при этом общие требования этого руководства сведены в конкретные документы и элементы управления проектом.

1 — Руководство РМВоК, например, требует наличия расписания проекта и иерархической структуры работ проекта. Технология корпоративного внедрения сводит эти два элемента в один документ — «План-график проекта». Следует отметить, что ГОСТ Р 54869-2011 «Проектный менеджмент. Требования к управлению проектом» также требует наличия одного документа — «Базового плана проекта».

Кроме того, общая методология всегда требует адаптации под конкретную организацию и конкретный проект, и ТКВ — не исключение. В результате анализа рисков конкретного проекта создания ИС, понимания того, какие группы рисков не несут серьезных угроз данному проекту, руководитель проекта может изменить:

- состав фаз базового жизненного цикла проекта<sup>1</sup>;
- состав выходной продукции в фазах жизненного цикла;
- роли, задействованные в рамках фазы, и выполняемую ими деятельность.

## Жизненный цикл проекта

Основой ТКВ является жизненный цикл проекта. Его цель — противодействие отрицательным особенностям ИТ-проектов и планомерное снижение определенных типов рисков в ходе выполнения работ

1 — Если не требуется вносить изменения в архитектуру типового решения, и технологические ограничения по производительности и масштабируемости системы маловероятны, можно убрать из жизненного цикла проекта фазу 2 «Архитектура системы».



**Рис. 2.** Источники технологии корпоративного внедрения и ее адаптация к условиям конкретного проекта создания ИС

по проекту. Базовый жизненный цикл проекта в ТКВ состоит из шести фаз:

1. Фаза 0. Инициация проекта.
2. Фаза 1. Концептуальное проектирование.
3. Фаза 2. Архитектура системы.
4. Фаза 3. Рабочий проект (разработка).
5. Фаза 4. Опытная эксплуатация.
6. Фаза 5. Ввод в промышленную эксплуатацию.

Помимо фаз, проект может разделяться на несколько этапов. В рамках каждого из них разрабатывается и внедряется полностью работоспособный релиз создаваемой ИС. В проекте может быть несколько этапов в зависимости от того, сколько релизов информационной системы необходимо будет разработать и внедрить в ходе проекта. Для создания полностью работоспособного релиза системы в рамках каждого этапа должны выполняться четыре фазы: «Архитектура системы», «Рабочий проект (разработка)», «Опытная эксплуатация» и «Ввод в промышленную эксплуатацию». Кроме того, в отдельный этап выделено «Завершение проекта<sup>1</sup>». Базовый жизненный цикл проекта в ТКВ показан на рис. 3.

Базовый жизненный цикл проекта в ТКВ можно отнести к итерационным<sup>2</sup>, он объединяет преимущества каскадной и инкрементальной моделей жизненного цикла.

Именно за счет такого сочетания обеспечивается скорость, гибкость и одновременно удешевление (относительное)

создания ИС. По мнению авторов методологии, на данный момент для разработки бизнес-приложений итерационная модель является наиболее оптимальной, относительно быстрой, качественной и недорогой. Можно выделить четыре особенности предлагаемого методологией базового жизненного цикла:

1. Наличие нескольких этапов, в рамках которых проходит полный цикл построения ИС (проектирование-кодирование-тестирование) — это характерная черта инкрементальных моделей<sup>3</sup> создания систем. Существование этапов позволяет:
  - выпускать несколько релизов системы в рамках одного проекта, что важно для больших корпоративных проектов, где инициация

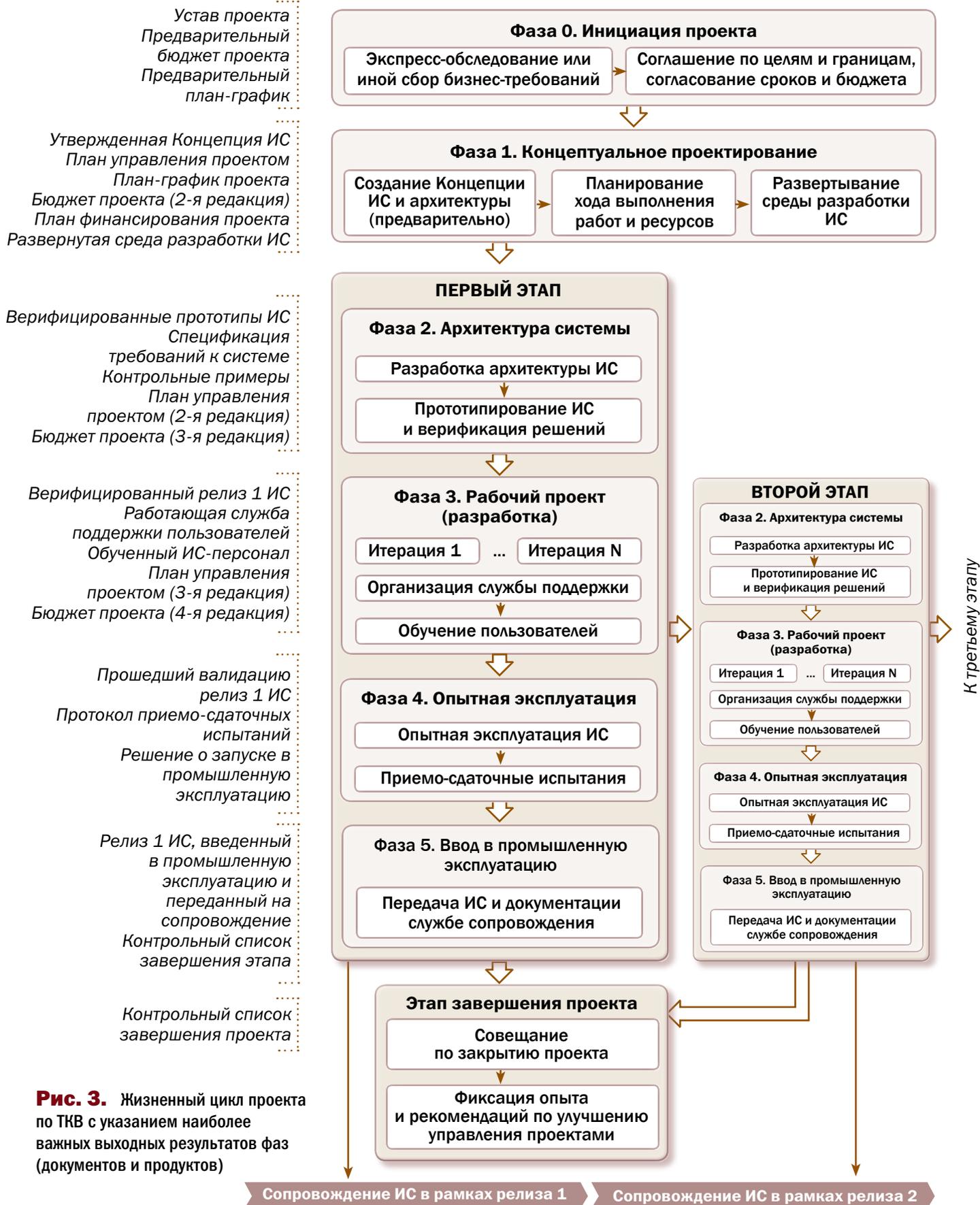
---

1 — Все стандарты и руководства выделяют завершение проекта в отдельную фазу жизненного цикла проекта. Однако, учитывая, что завершение проекта относится ко всем этапам, авторы методологии решили назвать эту стадию проекта этапом. При такой организации жизненного цикла это оправдано.

2 — Классический пример итерационного подхода к разработке — это методология разработки программного обеспечения, созданная компанией Rational Unified Process (RUP).

3 — Инкрементальная модель разработки предложена Б. Бозмом как усовершенствование каскадной (водопадной) модели. Она объединяет элементы последовательной каскадной модели с эволюционным подходом к созданию продукта

# ЭКСПЕРТНЫЙ СОВЕТ



**Рис. 3.** Жизненный цикл проекта по ТКВ с указанием наиболее важных выходных результатов фаз (документов и продуктов)

проекта — довольно длительный, трудоемкий процесс, и один проект с выпуском нескольких релизов ИС намного эффективнее, чем выполнение нескольких проектов для выпуска каждого релиза;

- уменьшать риск неопределенности требований к системе в начале ее создания (что случается весьма часто), постепенно и эволюционно подходить к разработке системы, учитывая новые появляющиеся требования и реализуя их на следующих этапах проекта;
- гибко учитывать неопределенность среды, в которой ведется проект, и реагировать на риски, обусловленные ситуацией в компании, в том числе на неготовность необходимых для успеха проекта активов: методологии, изменения организационной структуры и перераспределения полномочий, а также человеческого капитала.

2. Наличие нескольких итераций в рамках фазы разработки системы (Фаза 3. «Рабочий проект») характерно для итерационных моделей. В предлагаемом базовом жизненном цикле итерационность достигается в рамках одной фазы (а не за счет циклического возврата на предыдущую фазу), то есть целевые продукты фазы могут создаваться итерациями в рамках этой фазы. Например, на фазе «Архитектура системы» выпускается несколько про-

тотипов системы до тех пор, пока заказчик не утвердит ее архитектуру; на фазе «Рабочий проект» выпускаются несколько итераций системы, в которых постепенно наращивается функциональность. Это делается для того, чтобы снижать риски разработки. Хотя пока нет полностью работающей системы, но есть промежуточный работающий вариант, который можно запустить и протестировать, а заодно начать писать документацию, обучать пользователей и т. д.

3. Вынесение фазы «Концептуальное проектирование» за рамки этапов и итераций. На фазе «Концептуальное проектирование» фиксируются границы и рамки проекта. Эта фаза определяет общий вид системы. Во многих случаях цели, границы и рамки проекта в общих чертах понятны. И хотя документ «Концепция ИС», который является выходным продуктом этой фазы, может дорабатываться на последующих фазах, его существенное изменение требуется нечасто.

Если все же неопределенность требований к системе велика и в ходе проекта выясняется, что нужно изменять границы и рамки проекта (а значит, существенно менять сроки и бюджет проекта), то тогда запускаются механизмы управления изменениями, которые, в свою очередь, инициируют все необходимые работы, включая

дополнительное интервьюирование, сбор требований, изменение концепции, и соответствующие работы входят в следующий этап создания ИС.

4. Параллельное выполнение разработки системы и поддержка уже работающей версии. В случае наличия нескольких этапов разработки происходит параллельная разработка следующего релиза ИС и поддержка предыдущего релиза, уже введенного в промышленную эксплуатацию. Это требует грамотного распределения ресурсов службы эксплуатации и поддержки пользователей, учитывая, что ввод в эксплуатацию каждого релиза осуществляется в основном ее силами.

Отметим, что на фазе «Опытная эксплуатация» итерации не предусмотрены. По-

иск и устранение несоответствий системы требованиям к ней выполняются на фазе разработки («Рабочий проект») через верификацию (тестирование) и устранения дефектов. Изменения, вносимые в систему на фазе «Опытной эксплуатации», должны быть в основном косметическими — на приемо-сдаточных испытаниях уже поздно что-либо существенно менять. Основная цель фазы опытной эксплуатации заключается в том, чтобы заказчик убедился, что система работает так, как надо. Если на фазе «Опытная эксплуатация» выявится много несоответствий требованиям к системе, это означает, что команда проекта плохо работала, плохо управляла наиболее значимыми рисками проекта, и в таком случае надо ставить вопрос о ее компетентности или закрытии проекта.

