



KAPITEL 3 / CHAPTER 3³
**THE PROJECT MANAGEMENT DATA MODEL FOR INFORMATION
SYSTEMS DEVELOPMENT**

*МОДЕЛЬ ДАННЫХ УПРАВЛЕНИЯ ПРОЕКТАМИ РАЗРАБОТКИ
ИНФОРМАЦИОННЫХ СИСТЕМ*

DOI: 10.30890/2709-2313.2022-09-01-002

Введение

Развитие сектора информационных систем управления, имеющего большое значение с экономической и стратегической точки зрения, во многом зависит от успешного управления различными проектами в этой отрасли. Регулярные опросы показывают, что у ИТ проектов по-прежнему высокий процент неудачного завершения. Так, исследования, проводимые компанией Standish Group [1] с целью сделать отрасль более успешной, свидетельствуют, что ИТ проекты, выполненные в 2015 г., имели показатель успеха 29% по сравнению с 29% в 2011 г. и 28% в 2000 г. С другой стороны, в 2015 г. 52% проектов были признаны частично успешными (т.е., выполненными с опозданием, превышением бюджета и/или с меньшими, чем требуемые возможности и функции), в то время как, 19% оказались неудачными (отменены до завершения или были выполнены и никогда не использовались).

Конечно, хорошо подобранные лучшие методы управления проектами, методологии разработки информационных систем, методы и инструменты разработки способны как-то уменьшить риски. Однако постоянно возрастающая сложность ИТ проектов и среды, а также сокращение времени доставки продукта и низкие показатели успеха проектов свидетельствуют о возрастании потребности на изыскание новых и совершенствование существующих подходов, методов и средств для ведения разработок и управления проектами.

В качестве одного из эффективных способов решения рассматриваемой проблемы может быть рассмотрено создание инструментов управления проектами разработки информационных систем, которые на системном уровне интегрировали бы в себе достижения и тренды этой междисциплинарной области и учитывали специфические знания ИТ отрасли, существенно отличающие ее от традиционных ресурсоемких отраслей.

³ *Authors: Sukhomlinov A.I.*



Целью настоящей работы является разработка модели данных управления проектами разработки информационных систем, обеспечивающей интеграцию современных специфических знаний ИТ отрасли в область автоматизации проектного менеджмента, и предоставление, таким образом, возможности направленного выбора используемых методов и инструментов процессам управления ИТ проектами.

3.1. Концепции разработки информационных систем

Информационная система - это система, которая собирает, хранит, обрабатывает и доставляет информацию, относящуюся к организации (или обществу), таким образом, чтобы эта информация была доступна и полезна для тех, кто желает ее использовать, включая менеджеров, сотрудников, клиентов и граждан. Информационная система – это социальная система, поддерживаемая информационно-коммуникационными технологиями [1].

Деятельность, связанная с разработкой информационных систем, существует так долго, сколько существуют компьютеры. Выполненные проекты, накопленный опыт, теоретические предположения и обобщения эволюционно привели к возникновению целого ряда концепций, которые по замыслу их авторов должны были способствовать целенаправленному и эффективному управлению для придания создаваемым системам необходимых свойств, объективно требуемых условиями среды, в которой действуют предприятия. Такие концепции как жизненный цикл, методологии разработки, инженерия методов, фреймворки интегрированной архитектуры, инструменты архитектуры предприятия и подобные им другие получили сегодня широкую известность и все более интенсивно используются в современном профессиональном сообществе разработчиков информационных систем.

Жизненный цикл разработки информационных систем (System Development Life Cycle, SDLC), широко использовавшийся в 1970-х, представляет собой подход разработки информационных систем. Он включает в себя фазы, а также процедуры, задачи, правила, методы, принципы, документацию, средства и учебные программы, хотя точно не определяет какие. Предложенная им водопадная модель состоит из этапов разработки, которые должны выполняться последовательно. При этом выходы каждого



этапа «втекают» на вход следующего этапа и представляют собой аддитивные требования к создаваемому продукту (информационной системе), которые должны быть учтены и которым должен соответствовать, выход, создаваемый на последующем этапе разработки. Жизненный цикл разработки систем был хорошо испытан и проверен временем.

Методологии разработки систем (System Development Methodologies) появились в результате эволюции концепции «жизненный цикл разработки систем». Они возникли и развиваются из-за объективной необходимости их применения: улучшения конечного продукта (разрабатываемой системы), улучшения и стандартизации процесса разработки. [3]. Существует множество определений методологий разработки систем, однако наиболее полезное из них было сформулировано Британским обществом по вычислительной технике: «Методология - это рекомендованный набор философских подходов, фаз, процедур, правил, методов, средств, документации, менеджмента и обучения для разработчиков информационных систем» [3].

Методология разработки является более широким и комплексным понятием, чем подход. Некоторые методологии направлены на улучшение водопадной модели, другие предлагают новые альтернативные подходы. Альтернативные подходы можно классифицировать в рамках ряда широких тем – системные, стратегические, участвующие, прототипирующие, структурированные, основанные на данных, объектно-ориентированные, разрастающаяся или эволюционная разработка, ситуационный подход и др. [4]. К настоящему времени в мире создано более 1000 брендовых методологий и их количество продолжает увеличиваться.

Методологии, развивая принципы, заложенные в подходах, расширяют область этапов разработки, распространяясь, например, в область стратегии и планирования, заполняют разрывы в этапах, освещая более подробно весь спектр необходимой деятельности разработки, направляют разработку системы на получение предприятием новых конкурентных преимуществ. Распространение методологий в область стратегии и планирования обеспечивает согласованность между стратегией предприятия и его информационной системой и оказывает влияние на определение бизнес-стратегии предприятия за счет выявленных новых возможностей информационных систем. Кроме того, методологии порождают и вносят в область действия этапов конкретные, в том числе и новые, методы



моделирования, а также средства и инструменты автоматизации разработки. Так, например, методология «Информационная инженерия» [5] расширяясь в направлении корпоративного менеджмента или планирования, рассматривает анализ бизнес-целей организации, ее главных функций и информационных потребностей и включает этот этап в качестве начального при разработке информационной системы. Более того, заполняя разрывы в этапах, эта методология охватывает жизненный цикл разработки от этапа «стратегии» до «сопровождения» информационной системы, созданной на ее основе. Средство CASE CA GEN, являющееся программным воплощением этой методологии [6], обеспечивает автоматизацию проведения разработок систем и включает в себя весь необходимый набор методов и инструментов. Интегрированный репозиторий средства CA GEN обеспечивает интероперабельность требований и результатов, вырабатываемых на шагах и этапах разработки систем.

Инженерия методов (Method Engineering) - это инженерная дисциплина, предназначенная для проектирования, конструирования и адаптации методов, приемов и инструментов для разработки информационных систем [7]. Термин первоначально был введен в машиностроении для описания конструирования методов выполнения работы на заводах. Инженерия метода фокусируется не на приобретении готового метода у какого-либо поставщика, а на внутреннем построении методологического подхода для конкретной организации и/или конкретного проекта. Инженерия методов продвигает идею построения методологий для разработки информационных систем путем выбора и сборки фрагментов методов из репозитория. Этот репозиторий сначала необходимо заполнить автономными фрагментами, полученными из лучшего отраслевого опыта и совместимыми с данной метамоделью. Источник этих хранимых фрагментов не критичен для использования практикующими разработчиками. Фрагменты могут быть «вырезаны» из других, ранее существовавших методов, например, созданы из стандартизированной метамоделли.

Ситуационная инженерия методов (Situational Method Engineering) пытается взять лучшее из двух миров – ситуационного подхода и подхода инженерии методов [8]. Она обеспечивает построение методов из существующих, проверенных фрагментов. Создаваемые методы адаптированы к ситуации, в которой применяются. То есть, подход старается сохранить как преимущества стандартизации путем определения и повторного использования фрагментов метода, так и повысить гибкость, предоставляя средства для



настройки метода в соответствии с потребностями конкретного проекта.

Ситуационная разработка систем (Situational System Development) фактически отвергает формализованные методологии. Это можно охарактеризовать как возврат к периоду, когда методологии разработки информационных систем еще не существовали [4]. Подход состоит в использовании чего бы то ни было, что понимают разработчики, и что, они считают, будет работать. Он основан на навыках и опыте разработчиков и во многом опирается на них. Ситуационный подход предоставляет структуру для помощи разработчикам, но предполагает, что инструменты и методы, которые будут использованы (или адаптированы), или нет, зависят от ситуации. Ситуации могут различаться в зависимости, например, от типа проекта и его целей, организации и ее среды, пользователей и разработчиков, а также их соответствующих навыков. Тип проекта также может отличаться по своей цели, сложности, структурированности и степени важности, предполагаемому сроку действия проекта или его потенциальному воздействию.

3.2. Парадигмы управления проектами

Существующие многочисленные источники в целом определяют термин «проект» как нестандартную, разовую работу, состоящую, как правило, из набора взаимосвязанных задач, ограниченную по времени и бюджету для удовлетворения заданной потребности заказчика. В свою очередь, термином «управление проектами» обозначается применение знаний, навыков, инструментов и методов к проектной деятельности для удовлетворения требований проекта [2].

В отличие от методологий разработки информационных систем проектное управление существует с более давних времен, хотя таковым, как дисциплина, оно ранее никогда не называлось. До 1900 года проектами гражданского строительства обычно руководили творческие архитекторы, инженеры и сами мастера-строители. Как дисциплина управление проектами развилось из нескольких областей применения, включая гражданское строительство, машиностроение и тяжелую оборонную деятельность.

Отцом управления проектами в США считается Генри Гант. Он был инженером-механиком и консультантом по менеджменту, наиболее известен в



области управления как создатель «диаграммы Ганта», которая до сих пор считается важным инструментом управления. Его диаграммы, разработанные примерно в 1910–1915 годах, использовались в крупных проектах во время первой мировой войны. Пионером научного менеджмента в Восточной Европе, является польский инженер и экономист Кароль Адамецкий. Он является создателем теории, определяющую основу для регулирования планирования и контроля командной работы на производстве. Для этого он разработал графическую систему для одновременного представления нескольких сложных процессов, чтобы согласовать большое количество производственных операций. Графики указывали время начала и окончания действий и необходимые ресурсы для производства, имели различные формы и были адаптированы для различных промышленных применений. Первое применение его теории было реализовано в 1896 г., а результаты впервые доложены Обществу русских инженеров в 1903 г. [9].

С появлением и развитием области информационных систем управления, она, как и другие социальные виды деятельности, включая строительство, машиностроение и тяжелую оборонную промышленность, также стала сферой применения дисциплины управления проектами. Таким образом, термин «проект разработки информационных систем управления» может быть определен как «нестандартная, разовая работа, ограниченная по времени и бюджету, связанная с применением информационных и коммуникационных технологий людьми в организации, для удовлетворения заданных потребностей заказчика» [2].

Основные концептуальные термины парадигм управления проектами, отчасти, подобны терминам из области методологий разработки информационных систем. Жизненный цикл управления проектами, подход к управлению проектами, фреймворки управления проектами, методологии управления проектами, информационные системы управления проектами, управления проектами информационных систем - все эти термины используются в литературе по проектному управлению. Однако их толкование среди профессионалов не является однозначным, и это вводит в заблуждение технических специалистов по разработке информационных систем.

Как определяется в [10], жизненный цикл проекта представляет собой набор, иногда перекрывающихся, фаз проекта, название и количество которых определяется потребностями управления и контроля организации или



организации, участвующей в проекте, характером самого проекта и областью его применения. Жизненный цикл может быть задокументирован с помощью методологии. Жизненный цикл проекта может определяться или формироваться уникальными аспектами используемой организации, отрасли или технологии. Хотя у каждого проекта есть определенное начало и определенный конец, конкретные результаты и действия, которые происходят между ними, будут сильно различаться в зависимости от проекта. Жизненный цикл обеспечивает базовую основу для управления проектом, независимо от конкретной выполняемой работы.

Применяемая сегодня категоризация принципов и руководств проектного управления, определяющих, как осуществляется управление конкретным проектом, подразделяет его на традиционное (или классическое) и гибкое управление.

Традиционный подход к проектному управлению основан на идее, что проекты относительно просты, предсказуемы и линейны с четко определенными границами, что позволяет легко планировать в деталях и следовать этому плану без особых изменений. Конечной целью традиционного подхода к управлению проектами является оптимизация и эффективность в соответствии с первоначальным подробным планом проекта или, говоря обычным образом, завершение проекта в рамках запланированного времени, бюджета и объема. Традиционное проектное управление обычно выделяет последовательные этапы процессов управления проектом. Количество этапов управления проектами в различных руководствах варьируется. Так, широко известный стандарт PMBOK Института управления проектами (Project Management Institute, США) выделяет пять групп процессов или этапов управления: инициация, планирование, выполнение, мониторинг/контроль и завершение проекта [10]. В свою очередь, методология/подход PRINCE2 (AXELOS Limited, Великобритания) [11] определяет семь проектных управленческих процессов (этапов): управление проектом, инициирование проекта, контроль этапов, управление созданием продукта, управление границами стадий, закрытие проекта. Достоинством и преимуществом традиционного подхода является его надежность, обеспечиваемая жесткостью его структуры и единообразием [12].

Гибкий (Agile) подход к управлению проектами возник как реакция на основной недостаток традиционного подхода, а также в связи с



распространением концепции управления проектами на область программной инженерии и разработки программного обеспечения. Фактически он появился и развивается совместно с новыми подходами разработки программного обеспечения. Основным свойством подхода является то, что он обладает свойством адаптируемости к изменениям, как в течение жизненного цикла проекта, так и от проекта к проекту. Идея гибкого подхода заключается в том, что он, прежде всего, предлагает получение конечного результата за счет периодических изменений в ходе проекта, которые осуществляются итеративно. При этом каждая итерация, предпочтительно короткая, состоит из всех фаз разработки. На сегодняшний день описано не менее девяти различных методов Agile, включая Scrum, Crystal, Kanban, Extreme Programming, Adaptive Software Development, Agile Modeling, Dynamic Systems Development Method, Feature Driven Development и Internet Speed Development, которые различные источники относят к категории фреймворков, а иногда даже методологий.

Невозможно однозначно утверждать, что один из рассмотренных подходов лучше другого [12]. Выбор подхода зависит от особенностей проекта. Иногда необходимо использовать оба подхода. Традиционный подход больше подходит для проектов с четкими первоначальными требованиями пользователей и ясными целями и поэтому с очень низким уровнем неопределенности. Такие проекты имеют низкую скорость изменения требований, не требуют активного вовлечения пользователей. Он подходит для более крупных проектов и ситуации, когда требуется составление формальной документации на любом шаге проекта, и/или критичность системы является одной из ключевых характеристик проекта, а также, когда последствия отказа системы могут быть очень серьезными. В свою очередь, гибкий подход к управлению проектами предназначен, прежде всего, для творческих, инновационных проектов, таких как исследовательские проекты или проекты по разработке новых инновационных продуктов или даже проекты по улучшению процессов. Все такие проекты характеризуются высоким уровнем неопределенности, неясными целями проекта или неполными и непредсказуемыми запросами, в отношении которых можно предположить, что они будут существенно изменены в ходе проекта. Из-за постоянных запросов на изменение проекты организованы итеративно, нелинейно, с частыми изменениями и обновлениями плана проекта и требуют тесного и частого сотрудничества с конечным пользователем в ходе проекта. Типичный agile-проект представляет собой небольшой автономный проект разработки



программного обеспечения, чаще всего в рамках одной организации и обычно с упором на пользовательский интерфейс. Гибкие проекты не делают акцент на обширной документации, поэтому знания о проекте, в основном, неявны.

Институт управления проектами PMI определяет термин методология в широком его смысле как систему практик, методов, процедур и правил, используемых теми, кто работает в дисциплине [10]. Для случая применительно к управлению проектами используется термин методология управления проектами, который определяется как набор руководств и принципов, которые можно адаптировать и применять к конкретной ситуации. Здесь руководящие принципы могут быть такими простыми, как список задач, или это может быть особый подход к проекту с определенными инструментами и методами [12]. Несмотря на то, что и метод, и методология управления проектами могут считаться синонимами, поскольку оба основаны на базовом подходе и имеют определенные принципы и правила, обычно считается, что методология управления проектами включает в себя подробные методы и инструменты. Существующие современные руководства проектного управления в полной мере не могут считаться методологиями, их, скорее, следует относить к фреймворкам, или методам.

Как определяет [10], информационная система управления проектами - это информационная система, состоящая из инструментов и методов, используемых для сбора, интеграции и распространения результатов процессов управления проектами. Она используется для поддержки всех аспектов проекта от инициации до закрытия проекта и может включать как ручные, так и автоматизированные системы [10]. Среди коммерческих брендовых информационных систем традиционного управления проектами следует отметить Oracle Primavera, PM.soft, TILOS, Powerproject, Deltek Open Plan, Spider Project Professional и др. Наиболее известный и мощный из них программный продукт Oracle Primavera способен обеспечить потребности организаций, которые управляют большим количеством проектов одновременно, а также поддержку управления проектными группами в разных местах и на разных уровнях организации [13]. Кроме систем рассмотренного класса, на рынке также существует большое количество недорогих и более простых программных продуктов, реализующих как классический, так и гибкий подход. Из этих продуктов на автоматизированную поддержку гибкого подхода управления проектами ориентированы такие, как YouTrack, MeisterTask,



Metatask, Jira, Tagetprocess и др.

3.3. Требования к возможностям систем управления ИТ проектами

Учитывая рассмотренные концептуальные положения в области проектного управления и разработки информационной систем как специфической отрасли современной ИТ индустрии, можно выделить их особенности и сформулировать требования, которые в целом помогут определить требования к архитектуре информационной системы управления проектами разработки информационных систем предприятий.

1. Дисциплина «Управление проектами» применима к любому виду деятельности из всего ее разнообразия, существующего в социальной среде. В области информационных систем эта дисциплина может применяться к таким различающимся видам деятельности как разработка информационных систем, программная инженерия, реализация пакета, улучшение системы, консалтинг и бизнес-анализ, миграция систем, реализация инфраструктуры, аутсорсинг (и инсорсинг), аварийное восстановление и др. [2].

2. В современном проектном управлении рассматривается два подхода – классический и гибкий подходы. Ориентированный в большей степени на применение в области информационных технологиях гибкий (agile) подход управления проектами больше соответствует требованиям применения при создании небольших автономных программных продуктов с упором на пользовательский интерфейс.

3. Область разработки информационных систем и программная инженерия - это два существенно различающиеся вида деятельности. Программная инженерия рассматривает создание относительно автономного программного продукта, применение которого ограничено пользователями одного определенного типа. В отличие от нее разработка информационной системы акцентирует свое внимание на создании целевого продукта, соответствующего уникальным требованиям каждого предприятия и предназначенного для поддержки множества различающихся пользователей, деятельность каждого из которых направлена на достижение общей цели предприятия.

4. Создание программного обеспечения часто предполагает его массовое тиражирование, в противоположность этому тиражирование разработанной для



предприятия информационной системы не имеет практического смысла из-за уникальности его требований.

5. Управление проектами и сфера его применения «разработка информационных систем», в рассматриваемом случае представляют две пересекающиеся взаимосвязанные дисциплины, поэтому проектное управление должно активно учитывать специфику данной отрасли.

6. В области разработки информационных систем, несмотря на ее более позднее появление, проведены многочисленные исследования, получены важные результаты и накоплен опыт их практического применения. В том числе, возникли новые дисциплины, такие как «инженерия методов», «ситуационная инженерия методов» и «ситуационная разработка систем», улучшающие качество продукта и процесса, а также повышающие гибкость проведения разработок и управления такими проектами.

7. Существующие подходы гибкого управления проектами мало подходят для их распространения на область проектов информационных систем, так как они по своей сути поддерживают восходящее направление разработки и, тем самым, не способны обеспечить создание информационной системы управления, направленной на достижение предприятием поставленной цели.

8. Данные управления проектами разработки информационных систем должны, в основном, охватывать области операционного, технологического и производственного процесса предприятия, трудовых и инструментальных ресурсов, клиентов и их заказов/проектов. Они также должны быть релевантны требованиям возможных вариаций стратегий и процессов управления проектами на предприятии. С точки зрения управления предприятием в целом эти данные фактически являются подцепочкой цифрового следа деятельности всего ИТ предприятия.

9. Кардинальное улучшение качества проектного управления может быть достигнуто путем интеграции информационной системы управления проектами с информационной системой ИТ предприятия, а также со средствами workflow-технологий для осуществления активной коммуникации процессов и стейкхолдеров.



Особенность этой структуры состоит в использовании двух рекурсивных связей. За счет этого она обладает способностью представлять любую глубину вложенности и последовательность активации элементарных действий процесса разработки (фаз/шагов разработки). Структура позволяет наполнять и хранить в себе неограниченное количество методологий и повторно использовать любую из них по выбору при выполнении того или иного проекта. Структура МЕТОДОЛОГИЯ через неполную категоризацию представляет для каждой методологии корневые узлы, выполняющие роль идентификаторов и точек входа в каждую из методологий.

Каждое из элементарных действий методологии предполагает использование некоторого метода, который должен быть применен в ходе его выполнения, а также содержание и форму полученного при этом результата, который поступает на вход последующего элементарного действия методологии. Таким результатом/входом может быть, например, контекстная модель данных, модель данных в ключах, полностью определенная модель данных, модель потока диалога, функциональная декомпозиция и т.п. Структура МЕТОД отражает все методы, которые могут быть применены при разработке информационных систем. Применение метода может осуществляться с использованием программных пакетов средств CASE, все множество которых может быть представлено в структуре ИНСТРУМЕНТ. Модель связывает между собой структуры МЕТОДОЛОГИЯ/ФАЗА, МЕТОД и ИНСТРУМЕНТ, показывая тем самым возможности выбора методов и инструментов для каждого элементарного действия конкретной методологии.

Другая группа структур модели предназначена для представления данных о трудовых ресурсах и их организации. Одна из ключевых структур этой группы РАБОТНИК соответствует множеству всех работников ИТ компании. Модель фиксирует квалификацию каждого отдельного работника средствами структур КВАЛИФИКАЦИЯ, ИНСТРУМЕНТ, СЕРТИФИКАТ и соответствующих связей. Административно все работники ИТ компании организованы в виде структурных подразделений по профессиональному или иному признаку. Структура СТРУКТУРНОЕ_ПОДРАЗДЕЛЕНИЕ представляет все множество структурных подразделений ИТ компании. Рекурсивная связь «Являться_частью» устанавливает административную иерархическую связь линейной подчиненности структурных подразделений компании. Дополнительно к этому соответствующие связи устанавливают характер



принадлежности работников структуры РАБОТНИК к структуре СТРУКТУРНОЕ_ПОДРАЗДЕЛЕНИЕ. Структура ПРОЕКТНАЯ_ГРУППА входит в состав группы структур, отражающих проектную организацию работников компании, являющихся исполнителями проектов. Она позволяет идентифицировать и зафиксировать создание проектной группы, период ее существования и ее связь с проектом, для выполнения которого она создается. Структура ЗАКРЕПЛЕНИЕ и соответствующие связи закрепляют требуемых конкретных работников за проектными группами и назначают им на установленный период времени роли участия в проекте из перечня, хранимого в структуре РОЛЬ_В_ПРОЕКТЕ.

Группа структур ЗАКАЗЧИК и ПРОЕКТ обеспечивают представление данных о заказчиках и заказываемых ими проектах. Она позволяют идентифицировать и фиксировать общие данные, такие как дату открытия проекта, срок его выполнения, стоимость, фактическую дату начала и окончания выполнения проекта. Структура ПРОЕКТ имеет однозначную связь со структурой ПРОЕКТНАЯ_ГРУППА, определяющей в итоге всех исполнителей каждого конкретного проекта, а также со структурой МЕТОДОЛОГИЯ, определяющей также однозначно методологию, выбранную для применения в конкретном проекте.

Другая важная группа, связанная со структурой РАБОТА, предназначена для размещения данных о структуре производства работ (производственном процессе) ИТ компании. Она представляет всю совокупность данных предприятия о технологических процессах выполнения каждого отдельно взятого проекта. При этом каждая единица планирования и учета (планово-учетная единица) в структуре РАБОТА идентифицируется атрибутами «ID_проекта» и «Работа.ID_фазы», где последний атрибут однозначно определяется соответствующей «фазой» методологии. Каждой планово-учетной единице структура ставит в соответствие требуемый технический и трудовой ресурс, конкретных исполнителей, плановое и фактическое время начала и окончания выполнения работы.

Рассмотренная модель данных способна обеспечить процессы управления проектами, такие как инициирование, планирование, выполнение, контроль и закрытие проекта и др., а также операционные, технологические и производственные процессы для всех проектов ИТ предприятия в целом.

Инициирование проекта сопровождается занесением данных в структуры



ЗАКАЗЧИК, ПРОЕКТ И ПРОЕКТНАЯ ГРУППА. При этом атрибуты даты начала и даты окончания выполнения проекта остаются незаполненными.

Планирование проекта осуществляется параллельно с планированием технологического процесса и технологических операций. При этом осуществляется выбор, адаптация или инженерия нового метода/методологии с одновременным заполнением в общем случае структур МЕТОДОЛОГИЯ, МЕТОДОЛОГИЯ/ФАЗА, ПРЕДШЕСТВОВАНИЕ, МЕТОД, ИНСТРУМЕНТ, ТЕХНОЛОГИЯ_РЕАЛИЗАЦИИ, СРЕДСТВО_CASE, МЕТОД_ПОДДЕРЖКИ [7;8;14]. Одновременно устанавливаются ассоциации между структурами МЕТОДОЛОГИЯ и ПРОЕКТ, а также наполняется данными структура РАБОТА. Кроме того, осуществляется выбор технического ресурса, определение трудоемкости для каждой единицы работы и, соответственно, заполняются структуры ТЕХНИЧЕСКИЙ_РЕСУРС и ТРЕБОВАНИЯ_К_ТРУДУ проекта.

Процесс «выполнение работ» связан с периодическим занесением данных о фактическом и плановом времени начала и окончания работ, а также данных об исполнителях, определяемых средствами оперативного планирования или перепланирования по результатам контроля. При этом задействуются структуры РАБОТА, ИСПОЛНИТЕЛЬ и ЗАКРЕПЛЕНИЕ. Планирование или перепланирование операций ИТ предприятия может выполняться программным модулем с применением современных методов оптимизации.

Процесс контроля основывается на анализе данных структуры РАБОТА и состоит в принятии решения об успешности хода выполнения работ или о перераспределении ресурса.

Закрытие проекта сопровождается присваиванием соответствующих значений атрибутам «Дата_окончания» структур ПРОЕКТ и ПРОЕКТНАЯ_ГРУППА. При этом ни одно из данных, поступивших в структуры в ходе выполнения проекта, не разрушается, оставляя, таким образом, цифровой след хода выполнения проекта и обеспечивая тем самым возможность многократного проведения аналитики в дальнейшем.

Предложенная модель данных может быть использована в качестве основы как для создания структур данных специализированных информационных систем управления проектами разработки систем предприятий, так и для расширения структуры баз данных действующих интегрированных систем управления ИТ предприятий.



Выводы

В работе рассмотрены и обсуждены концепции разработки информационных систем и проектного управления - подходы, методы, методологии, а также инструменты, основанные на этих концепциях. Анализ концепций показал наличие между ними существенных связей. При этом было выявлено, что подход и структура процессов управления ИТ проектом, в основном, определяется методологией, применяемой при разработке системы. Рассмотрена возможность интеграции знаний разработки информационных систем в проектное управление и, соответственно, в программные средства его автоматизации. Предложено использовать спектр существующих подходов и методологий разработки информационных систем, в том числе, и гибкий подход, основанный на инженерии методов разработки, в качестве источника знаний. Разработана модель данных для размещения в определяемых ею структурах интегрированных знаний проектного управления и методологий разработки информационных систем. Приведено описание структур модели данных, а также правил их заполнения и использования процессами управления проектами. Результаты работы могут быть использованы в качестве основы для практического применения при усовершенствовании и автоматизации проектного управления в ИТ предприятиях.