

ARCHITECTURAL COMPONENT-ORIENTED APPROACH TO MANAGING IT PROJECTS

АРХИТЕКТУРНЫЙ КОМПОНЕНТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К УПРАВЛЕНИЮ ИТ-ПРОЕКТАМИ

*Maryna P. Chaikovska
Oleksandr A. Chaykovsky*

Abstract. The article deals with the features and problems of IT project management. It analyzes the modern methodologies of IT project management, their characteristics, limitations, and spheres of applicability. The advantages of an architectural component-oriented approach to managing IT projects based on the Kenevin complexity model are described. Examples of effective practical implementation of the approach within Microsoft and 1C corporations are given.

Key words: IT project, methodology of IT project management, life cycle model, complexity of IT project, Kenevin model.

Аннотация. Рассмотрены особенности и проблематика управления ИТ-проектами. Проанализированы современные методологии управления ИТ-проектами, их характеристики, ограничения, сферы применимости. Описаны преимущества архитектурного компонентно-ориентированного подхода к управлению ИТ-проектами на основе модели сложности Кеневин. Приведены примеры эффективной практической реализации подхода в деятельности корпораций Microsoft и 1С.

Ключевые слова: ИТ-проект, методология управления ИТ-проектами, модель жизненного цикла, сложность ИТ-проекта, модель Кеневин.

В современном мире информация становится ключевым ресурсом и продуктом общества, по своей значимости и масштабам, превосходящим материальную продукцию.

Формирование и становление информационного общества непосредственно связано с необходимостью решения информационных проблем трансформирующегося предприятия, с усилением роли информационных систем и технологий в менеджменте организаций.

Влияние информационных систем вызывает модификацию структуры организации, распределение полномочий и ресурсов, меняет характер труда; проявляется в повышении гибкости механизмов работы менеджеров, снижении транзакционных и управленческих затрат, сокращении количества организационных уровней управления.

Объективной реальностью становится смещение вектора локализации ключевых конкурентных преимуществ предприятия XXI века от параметров цена-качество в направлении вопросов оперативности управления бизнес-процессами, скорости изменения бизнес-моделей, гибкости и адаптации к условиям окружающей среды.

Конкурентоспособность современного предприятия непосредственно связано с наличием необходимой информационно-технологичной инфраструктуры; умением персонала ее эффективно использовать; обеспечением ее масштабирования и актуализации.

Трансформационные глобальные процессы информатизации общества обуславливают динамичный рост масштабов и разнообразия информационных задач предприятия, для решения которых необходима разработка эффективной методологии управления проектами информатизации (ИТ-проектами) в соответствии с целями и стратегиями организации.

В условиях современного бизнеса залогом успешности и конкурентоспособности предприятия компании является эффективное управление проектами информатизации (проектами создания, внедрения и развития информационных систем предприятия, развертывание ИТ-инфраструктуры) на базе использования инновационных методологий.

Эффективность реализации современных ИТ-проектов является критическим вопросом и залогом успешности в управлении информационными потоками на разных уровнях как экономических, так и экологических систем. Однако процесс разработки и внедрения информационной системы сегодня является сложным, длительным, затратным проектом, связанным с большими рисками, требует применения адекватных методологий управления.

Управление проектами заключается в осуществлении и доведении проекта до логического завершения путем организации и управления людьми, временем, издержками и ресурсами.

Управление проектом – это применение знаний, навыков, инструментов и методов к работам проекта для удовлетворения требований, предъявляемых к проекту⁶⁸.

Характерной особенностью современных ИТ-проектов является наличие изменений в проекте, которые могут касаться не только условий реализации проекта, но и самой цели проекта или его качественных характеристик.

Одной из главных задач, напрямую влияющих на эффективность проекта информатизации, является правильный выбор адекватной модели процесса и методологии внедрения как необходимого условия эффективного управления сроками, стоимостью и рисками ИТ-проектов, повышение управляемости и прогнозируемости проекта.

Вопросы управления ИТ-проектами чаще рассматривается в общем ключе методологии проектного менеджмента, однако они имеют ряд существенных различий, порождаемых спецификой задачи, многообразием и сложностью современных ИТ-проектов, наличием изменений в проекте, связанных не

⁶⁸ РМВОК. Руководство к Своду знаний по управлению проектами, 5-е изд., PMI, 2014., с. 33.

только с условиями реализации проекта, но и самой целью проекта и ее качественными характеристиками.

Различия в способах решения задачи требует особой методологии управления ИТ-проектами и проектными рисками. Существуют различные стандарты, как международные, отраслевые, так и разработки ведущих мировых ИТ-компаний, однако все они носят скорее рекомендательный характер, обобщая успешный опыт реализации ИТ-проектов.

Сегодня все попытки формализации и выработки единой эффективной методологии управления ИТ-проектами оказались несостоятельными, уникальность ИТ-проектов выдвигает на первый план вопрос выбора методологий с целью повышения эффективности их реализации и снижения проектных рисков.

Целью данного исследования является анализ особенностей, рисков современных ИТ-проектов, систематизация методологий управления ИТ-проектами, выработка рекомендаций по их применимости.

Актуальность темы усиливается такими характерными чертами современных проектов информатизации, как углубление специализации, динамизм требований пользователей, многообразием проектных рисков, а также активным развитием моделей, методов, методологий и программного инструментария проектов информатизации.

Статистика анализа успешности ИТ-проектов последние 10 лет с небольшими колебаниями устойчиво удерживается в диапазоне 20-30%. Процент успешных проектов вырос всего на 10%, а доля провальных проектов практически не изменилась.

В исследовании The Standish Group⁶⁹ классифицирует все существующие проекты на три группы: успешные, провальные и спорные.

Успешные проекты (Successfull) – проект выполнен в рамках тройного ограничения (все поставленные цели проекта реализованы в намеченное время и не превышают выделенный бюджет).

Провальные проекты (Failed) – проекты, не достигшие никаких результатов (деньги, затраченные впустую).

Спорные проекты (Challenged) – проекты, которые по стоимости обходились дороже, чем было запланировано или которые были завершены не вовремя (превышая сроки) или не завершены.

Исследования Standish Group, основанные на анализе 50000 проектов, показывают следующую статистику. Порядка 31,1% ИТ-проектов оказались “провальные” по тем или иным причинам и были остановлены, так и не достигнув никаких целей. 52,7% ИТ-проектов оказались “спорные”, поскольку столкнулись во время разработки или внедрения с проблемами, сказавшимися на качестве, бюджете и длительности проекта. Это повлияло на выполнение запланированных целей и привело к нежелательным результатам. Для проектов, завершенных с опозданием или аннулированных до завершения, среднее превышение бюджета составило 89%, а срок выполнения – на 122% .

⁶⁹ David Rubinstein (2017) Standish Group Report: There's Less Development Chaos Today.

По оценкам Gartner Group, в 2017-м расходы на информационные технологии в глобальном масштабе достигли \$3,46 трлн., что на 2,7% превышает показатели предыдущего года. Средняя стоимость завершённых проектов в сфере информационных технологий в 2014 году составила 189% от первоначальных оценок.⁷⁰

В Украине лишь 4% ИТ-проектов завершаются вовремя, что является одним из самых низких показателей в Европе. Лидерами по этому показателю в Европе являются: Швеция (44%), Швейцария (22%), Чехия (20%), Германия (19%) и Дания (16%).⁷¹

Для крупных проектов (стоимость человеческих ресурсов больше \$10 млн.) вероятность успеха составляет лишь 10%. В то время как проекты со стоимостью человеческих ресурсов менее \$1 млн. (малые) являются успешными с вероятностью 75%.

Основными проблемами ИТ-проектов являются: частые изменения спецификаций и искажение данных, полученных от заказчика. Это приводит к увеличению стоимости работ и получению продукта, который оказывается не актуальным для заказчика. Таким образом, на успех проекта напрямую влияет тщательность сбора и управления требованиями к проекту и продукту в процессе управления их содержанием

Классический проектный менеджмент строго привязан ко времени исполнения задач, как правило, заранее определённого на этапе планирования. Для реализации проектов в рамках данного подхода рекомендуются инструменты календарно-сетевое планирования: Диаграмма Ганта, PERT диаграммы, метод CPM, WBS.

Диаграмма Ганта показывает расписание проекта, основываясь на датах окончания и завершения задач. В неё вносятся задачи, их длительности и взаимосвязи, а затем высчитывается критический путь – самая длинная цепочка взаимосвязанных задач, определяющих длительность проекта. Проектам с жёсткими дедлайнами диаграмма Ганта помогает решить, когда лучше начинать те или иные задачи, чтобы сократить время реализации. А для проектов с сильными ресурсными ограничениями, диаграмма Ганта предоставляет возможность построить схему в форме событийной цепочки процессов для планирования ресурсов.

PERT (Project Evaluation and Review Technique) – способ анализа задач, необходимых для выполнения проекта; анализа времени, которое требуется для выполнения каждой отдельной задачи; определения минимального необходимого времени для выполнения всего проекта с учетом наличия неопределённости (возможность разработать рабочий график проекта без точного знания деталей и необходимого времени для всех его составляющих).

CPM (Critical Path Method) – метод критического пути, разработанный для контроля сроков выполнения проекта. Предпосылкой применения метода CPM

⁷⁰ Gartner Research, 2017.

⁷¹ Chaikovska M. Methodological bases of IT-Project management with simulation modelling tools, p. 55.

является предположение о том, что время выполнения каждой работы точно известно.

WBS (Work Breakdown Structure) – разбиение проекта на конкретные результаты, которые должны быть достигнуты для достижения целей проекта и их группировки.

Свод знаний по управлению проектами (PMBOK – The Project Management Book of Knowledge), рассматривает управление проектом сквозь призму 5 ключевых групп процессов (инициирование, планирование, выполнение, мониторинг, завершение) и 9 областей знаний (управление интеграцией, управление содержанием, управление временем, управление стоимостью, управление качеством, управление ресурсами, управление коммуникациями, управление рисками, управление закупками). PMBOK предлагает набор специализированных аспектов управления проектом и конкретных практик и инструментов управления проектами (таких как диаграмма Ганта, CPM, PERT, WBS).

ИТ-проекты можно рассматривать в виде общего термина “проект” но с некоторыми важными дополнениями, связанными, прежде всего, с тем, что ИТ-проект подразумевает собой деятельность, направленную на создание или использование информационных технологий.

В работах Фатрелла Г., Дональда Ф. Шафера, Расмуссона Дж., Томсетта Р., Арчибальда Р. Д. делается акцент на управление ИТ-проектами, однако больше с технической, чем с организационной точки зрения⁷².

ИТ-проекты представляют собой особый тип проектов, характеризующийся:

- неоднозначными потребностями пользователей ИС;
- с отсутствием взаимопонимания команды разработки и пользователей;
- отсутствием приоритетов, противоречивостью, неточностью требований;
- недостатком ресурсов, конфликтами за ресурсы, проблемами планирования;
- сжатыми, нереалистичными сроками;
- недостаточной детализацией планирования;
- слабо идентифицированными рисками.

С целью повышения качества реализации ИТ-проектов разработаны типовые методологии управления процессами разработки программного обеспечения:

SWEBOK (Software Engineering Body of Knowledge)⁷³ – открытый проект, реализуемый при поддержке международного комитета IEEE начиная с 1998 года, предлагает основные теоретические и практические знания, накопленные в этой отрасли. В нем определен набор знаний и рекомендуемые практики по управлению ИТ-проектами.

С 1990-го по 1995 год велась работа над международным стандартом, который должен был дать единое представление о процессах разработки

⁷² Chaikovska M. Adoptive models of management modern IT-projects, p. 115.

⁷³ Bourque P., Fairley R. E. SWEBOK V 3.0. Guide to the Software Engineering Body of Knowledge.

программного обеспечения. В результате был выпущен стандарт ISO/IEC 12207.⁷⁴ Согласно стандарту, жизненный цикл программы, программной системы, программного продукта включает в себя разработку, развертывание, поддержку и сопровождение. Если программный продукт не коробочный, а достаточно сложный, то его развертывание у клиентов, как правило, реализуется отдельными самостоятельными проектами внедрения.

К настоящему времени в мире существует целый ряд универсальных методологий:⁷⁵

- PMI (разработана международным некоммерческим институтом управления проектами, не привязана к предметной области).

- IPMA (разработана международной ассоциацией управления проектами, не привязана к предметной области).

- CMMI (Университет Карнеги-Мелона) – ориентирована на разработку ПО.

- Six Sigma (1986, Компания Motorola, концепция планирования для экономии ресурсов, повышения качества, снижения количества брака и проблем на основе 5-шагового процесса Define, Measure, Explore, Develop, Control с целью удовлетворение заказчика качеством продукта, которого можно добиться при помощи непрерывного процесса улучшения всех аспектов проекта, основанном на тщательном анализе показателей).

- PRINCE2 (Projects in Controlled Environments)⁷⁶ – структурированная процессно-ориентированная проектная методология, которая фокусируется на процессах верхнего уровня (управление, организация, контроль), а не на низших задачах (декомпозиция работ, разработка графиков). Одобрена правительством Великобритании в качестве стандарта управления проектами. Ориентирована на ИТ-проекты, разработана Центральным компьютерным и телекоммуникационным агентством Великобритании, зарегистрирована в Министерстве торговли Великобритании – Office of Government Commerce.

PRINCE2 концентрируется на управленческих сторонах проекта, выраженных в 7 принципах (общие правила управления проектами), 7 процессах (шаги продвижения по проектному циклу) и 7 темах проекта (контроль для достижения успеха проекта). Является гибридом классического подхода к проектному управлению и концентрации на качестве из 6 сигм.

- COBIT (Control Objectives for Information and related Technology) – стандарт управления и аудита в области информационных технологий (Планирование и Организация, Проектирование и Внедрение, Эксплуатация и Сопровождение, Мониторинг).

- SSADM (Structured Systems Analysis And Design Methodology) – с 1993 является национальным стандартом Великобритании для разработки информационных систем. Методология структурного системного анализа и проектирования, регламентирует начальные этапы разработки системы. В SSADM применяется нисходящий подход к построению интегрированных

⁷⁴ ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.

⁷⁵ Cockburn, A. Methodology per Project.

⁷⁶ PRINCE2® 2009 Edition A Pocket Guide.

функциональных, информационных и событийных моделей. Отличительной чертой SSADM является четкое выделение и поддержка соответствующими методиками так называемых “нефункциональных требований”.⁷⁷

- RUP (Rational Unified Process) – унифицированный процесс, разработанный компанией "Rational Software" в качестве дополнения к языку моделирования UML. В основе методологии RUP лежит "итеративный пошаговый подход", определяющий этапы жизненного цикла, контрольные точки, правила работ для каждого этапа. Этот подход обеспечивает большую гибкость при изменяющихся требованиях и тактических коррективах в бизнес-целях, что позволяет более эффективно и заблаговременно идентифицировать и снижать проектные риски.⁷⁸

- MSF (Microsoft Solutions Framework) – методология разработки программного обеспечения, опирается на практический опыт Microsoft и описывает управление людьми и рабочими процессами в процессе разработки решения.

- RAD (Rapid Application Development) – быстрая разработка приложений, используемая в проектах по разработке ПО. Основной целью является быстрое и качественное создание приложения; помогает улучшить показатели результативности проекта и повысить качество риск-менеджмента; выделяет 4 стадии проекта: планирование, пользовательское проектирование, быстрое конструирование, переключение; не подходит для масштабных IT проектов. Рекомендована для относительно небольших проектов, разрабатываемых для конкретного заказчика. Большое значение имеет опыт и профессионализм разработчиков.⁷⁹

- Crystal (Алистер Коберн 1997) – методология целесообразна для использования небольшими (7-8 членов группы) командами для локальных некритичных ИТ-проектов.

- FDD (Feature driven development) – разработка, управляемая функциональностью) фокусируется на пятишаговом подходе, который базируется на идентификации, разработке и внедрении функций. В FDD разрабатываются функции, которые решают определенные задачи, а потом они объединяются в один продукт. Эта методология тоже больше подходит для команды, состоящей из разработчиков.

- XP (Extreme Programming). В экстремальном программировании к принципам гибких методологий добавляются рекомендации улучшения работы с кодом (парное программирование, автоматическое тестирование, стандарты кодирования); методология разработана для команд программистов.⁸⁰

- Agile – класс гибких методологий, базирующихся на адаптивной разработке, ориентированность на людей и их взаимодействие, а не на процессы и средства. Характеризуются гибкостью, итеративностью, адаптивностью. Предлагает разбивать на небольшие управляемые пакеты

⁷⁷ Barry Boehm, et al. Software cost estimation with COCOMO II. Englewood Cliffs.

⁷⁸ Paulk, Mark C., and others, Capability Maturity Model for Software, Version 1.

⁷⁹ Kent Beck, and others, Manifesto for Agile Software Development.

⁸⁰ Jeffries, R., Beck, K., et al. Extreme Programming, as described on the web.

работ. Итеративно-инкрементальный подход к управлению проектами и продуктами, ориентированный на динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля. Работающий продукт важнее исчерпывающей документации; сотрудничество с заказчиком важнее согласования условий контракта; готовность к изменениям важнее следования первоначальному плану. Целесообразно для проектов с “открытым концом”.⁸¹

Существует множество методов, базирующихся на идеях Agile – Scrum, Kanban, Lean.

- Kanban (Toyota, 1953) метод управления разработкой, реализующий принцип точно в срок и способствующий равномерному распределению нагрузки между работниками. Инкремент продукта передаётся вперёд с этапа на этап, а в конце получается готовый к поставке элемент. Точный расчёт нагрузки на команду, правильная расстановка ограничений и концентрация на постоянном улучшении. Основная задача канбан — уменьшать количество “выполняющейся в данный момент работы”. В канбан оценки сроков на задачу опциональные или вообще их нет. Канбан — это инструмент визуализации процесса и ограничения выполнения одновременного количества задач, но его одного недостаточно для успешного выполнения проекта. Подходит для достаточно сплочённых команд с хорошей коммуникацией.⁸²

- Скрам – наиболее популярная гибкая методология, широко используемая зарубежными компаниями, такими как Yahoo!, PayPal, Google, GE для управления проектами. Согласно одиннадцатому ежегодному исследованию гибкой разработки VersionOne, методология скрам используется как метод гибкого управления проектом в несколько раз чаще, чем остальные методологии.⁸³ Скрам является самым структурированным из семейства Agile, сочетает в себе элементы классического процесса и идеи гибкого подхода к управлению проектами. Скрам – это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами, предоставлять заказчику продукт. Методология скрам нацелена на взаимодействие с заказчиком, и, несмотря на то, что команда разработчиков сама решает, какие задачи она будет выполнять в течение одной итерации, в данной методологии присутствует руководитель (скрам-мастер), который контролирует соблюдение процесса. Использование этой методологии дает возможность выявлять и устранять отклонения от желаемого результата на более ранних этапах разработки программного продукта. Чем меньше формализма, тем более гибко и эффективно можно работать, – это основной принцип данной методологии. Но это не означает, что формальных процессов не должно быть совсем, их должно быть достаточно для организации эффективного взаимодействия и управления проектом.

⁸¹ Cohn M. Succeeding with Agile: Software Development Using Scrum, p. 112.

⁸² Cobb G. Making Sense of Agile Project Management: Balancing Control and Agility, p. 54.

⁸³ The Scrum Guide.

Формальная часть скрам состоит из трех ролей, трех практик и трех основных документов.

- Lean Development (LD) – бережливая разработка ПО. Гибкая методология, предполагающая сохранение высокого морально-функционального состояния разработчиков. Основана на тезисе “Мыслить широко, делать мало, ошибаться быстро, учиться стремительно”. Позволяет параллельно выполнять несколько задач на разных этапах, что повышает гибкость и увеличивает скорость исполнения проектов; сочетает гибкость и структурированность, однако не рекомендуется для крупных и неоднородных проектов т.к. не предполагает чёткого рабочего процесса для реализации элементов проекта, что способствует растягиванию сроков проекта.⁸⁴

Почему же при таком разнообразии методологий и накапливаемом опыте управления ИТ-проектами процент успешных ИТ-проектов продолжает оставаться недостаточно оптимистичным? Не является столь высокое разнообразие избыточным и тавтологичным?

Проведенный анализ методологий позволяет выделить общие черты, ограничения и сформулировать сферы применимости соответствующей специфики ИТ-проектов.

Каждый ИТ-проект проходит путь с момента своего создания и до его завершения (когда информационная технология полностью создана или внедрена) через ряд промежуточных этапов, фаз, совокупность которых называют общим термином – жизненный цикл.

Свод знаний по управлению проектами (PMBoK) дает следующее определение понятию жизненный цикл проекта: “Жизненный цикл проекта – это набор, как правило, последовательных и иногда перекрывающихся фаз проекта, названия и количество которых определяются потребностями в управлении и контроле организации или организаций, вовлеченных в проект, характером самого проекта и его прикладной областью”.⁸⁵

Модель жизненного цикла разработки программного обеспечения – структура, которая определяет последовательность выполнения и взаимосвязей процессов, действий, заданий.

Эволюционный вектор развития методологий направлен в сторону увеличения гибкости каскадной модели ЖЦ (V-образная модель, итерационная модель, интерактивная, спиральная модель), за счет стандартизации, контроля и документирования.

Современным трендом от ведущих ИТ-вендорами является отход от универсальных методологий к архитектурному компонентно-ориентированному подходу на основе модели сложности Кеневина (Synefin framework).⁸⁶ Synefin framework позволяет категорировать ИТ-проекты по сложности (проекты упорядоченные простые, проекты упорядоченные сложные, проекты запутанные, проекты хаотичные) и двигаться в сторону повышения гибкости методологии или выбора классической методологии.

⁸⁴ M. Pomeroy-Huff, J. Mullaney, R. Cannon, M. Sebern, The Personal Software Process (PSP) Body of Knowledge.

⁸⁵ Paulk, Mark C., and others, (1993) Capability Maturity Model for Software.

⁸⁶ A Leader's Framework for Decision Making – Harvard Business Review.

Проекты упорядоченные простые. Характеризуются четкими взаимосвязями, в них реально найти эффективный способ достижения результата. Наиболее целесообразно использовать метод “Лучших практик” (Best practice) на базе каскадной модели жизненного цикла (waterfall).

Проекты упорядоченные сложные. Причинно-следственные связи есть, но уже не столь очевидны. Экспертные методы позволяют обнаружить взаимосвязи. Применим подход – “Хорошие практики”. Подходят методы проектного управления (PMI, Prince2, СММІ).

Проекты запутанные. Характеризуются запутанными и разнообразными связями, большим количеством участников проекта, сложными и неочевидными причинно-следственными связями, неоднозначными результатами. Применим итерационный метод. Практики нарабатываются в ходе проведения экспериментов. Целесообразны гибкие методологии Agile, Scrum, Lean.

Проекты хаотичные. Причинно-следственных связей нет. Невозможно делать никаких выводов. Из состояния хаоса есть два возможных выхода: введение в систему жестких ограничений и ее переход в упорядоченную и простую, либо быстрыми действиями, снижающими турбулентность, позволить системе перейти в состояние запутанной.

Примерами практического применения архитектурного подхода является обобщенный, наиболее успешный опыт групп разработки продуктов, опыт заказчиков и партнеров ИТ корпорации Microsoft (Microsoft Solutions Framework) и ИТ корпорации 1С. Факторы успеха интегрируются во взаимосвязанные модели, применение которых позволяет получить технологичные решения в контексте конкретного бизнеса.

Microsoft Solutions Framework (MSF)⁸⁷ – это гибкая модель, построенная на основе итеративной разработки, небюрократизированной проектной команды, нестандартных подходах к организационной структуре, распределению ответственности и принципам взаимодействия внутри команды. MSF включает модель проектной группы MSF (Team Model), определяющую основные роли, закрепленные за членами проектной группы, для того чтобы сделать проект удачным и модель процесса проектирования (Process Model). Process Model направлена на решение проблем спиральной модели, вводя понятие "вех" (точек синхронизации проектной группы и заказчика), укорачивая цикл проектирования с помощью механизма выпуска версий.

Архитектурный компонентно-ориентированный подход на основе модели Кенефина (Synefin framework) проявляется в разделении ИТ-проектов на проекты подготовки к внедрению информационной системы, проекты внедрения информационной системы; проекты по поддержке информационной системы внедренной на предприятии. Позволяет представить ИТ-проект в виде набора основных компетенций и компонент.

Состав компонент определен на основе “лучших практик”. Такой подход помогает идентифицировать возможности для улучшения и инноваций, в том

⁸⁷ Microsoft Solutions Framework (MSF).

числе провести укрупненный анализ принятых решений по внедрению процессов управления ИТ и связанными с ними организационными преобразованиями. Соответственно выделяются области деятельности и компонентные стандарты (табл. 1).

Таблица 1 – Компонентная архитектура подхода Microsoft

Компонент	Тип ИТ-проекта	Функции компоненты
MRF	подготовка к внедрению информационной системы	формализуются требования к ИТ и определяется объем проекта
MSF	внедрению информационной системы на предприятии	определяются основные вопросы разработки и развертывания ИТ-решения
MOF	по поддержке информационной системы внедренной на предприятии	вопросы эксплуатации информационной системы

Методология проектного управления, предлагаемая фирмой “1С”,⁸⁸ формируется на основе Технологий проектного управления для проектов различных типов и масштабов.

Для управления проектами внедрения программных продуктов на платформе 1С: Предприятие, фирма “1С”, предлагает три технологии (табл. 2):

1С: Технология Стандартного Внедрения (1С: ТСВ).

1С: Технология Быстрого Результата (1С: ТБР).

1С: Технология Корпоративного Внедрения (1С: ТКВ).

Таблица 2 – Компонентная архитектура подхода 1С

Компонент	Характеристики	Сфера применимости
1С:ТСВ	Технология управления проектами внедрения программных продуктов. Трехфазная модель (планирование, внедрение, регулярное сопровождение).	ориентирована на внедрение типовых решений; акцент на планирование и снижение уровня рисков работ по внедрению; требует активного вовлечения заказчика в проект
1С:ТБР	Технология управления проектами внедрения тиражных программных продуктов. Четырехфазная модель (инициация проекта, управление требованиями и ИТ-инфраструктурой, внедрение, передача на сопровождение).	быстрый результат достигается за счет снижения транзакционных издержек и высокоэффективных коммуникаций в проектной команде
1С:ТКВ	Технология управления проектами разработки ПО. Ориентирована на выполнение масштабных проектов с задействованием большого количества ресурсов и автоматизацией сразу нескольких функциональных областей	акцент на управление функциональными и нефункциональными требованиями; направлена на минимизацию рисков; требовательна к уровню квалификации руководителя проекта.

⁸⁸ 1С:<https://www.softcom.ua/ru/1c/>.

Сложные проекты могут включать в себя как интеграцию отдельных готовых компонентов информационной системы, так и построение на их основе специальных решений. Чем сложнее проект, тем более формализованного подхода он требует.

Масштаб проекта зависит не только от параметров объекта автоматизации, но и от исполняющей организации, степени модификации типового решения, уровня коммуникаций, скорости принятия решений, размера команды проекта, требований к формализации документации. В таблице 3 приведены результаты сравнительного анализа моделей по основным компонентам, выделены ограничения применимости.

Таблица 3 – Компоненты методологий в разрезе модели Кеневин

Модель	Этапы	Коммуникации	Документирование	Ограничения
Тяжелые	Формализованы по объему и сложности проекта. Акцент на стадии анализа и проектирования. Прогнозируемы. Фокусируются на детальном планировании.	Осуществляется через документирование. Требования к стабильности команды невысоки. Необходима существенная управленческая надстройка.	Разработка полного пакета документации на каждом этапе проектного цикла. Стандартизация и регламентация документооборота.	Специализация исполнителей высока. Процессы рассчитаны на среднюю квалификацию исполнителей
Легкие	Упрощенные стадии анализа и проектирования, основной упор на разработку функциональности, совмещение ролей.	Гибкие команды, меньше непроизводительных расходов, связанных с управлением проектом, рисками, изменениями, конфигурациями. Неформальные коммуникации	Нацелены на преодоление ожидаемой неполноты требований и их постоянного изменения	Эффективность сильно зависит от индивидуальных способностей, требуют более квалифицированной универсальной и стабильной команды. Объем и сложность выполняемых проектов ограничены.

В рамках управления проектом происходит постоянное взаимодействие участников проекта, к которым относятся инициатор, заказчик, менеджер проекта, инвестор и, непосредственно, команда проекта.

Любая методология базируется на модели процессов (модели жизненного цикла информационной системы), модели команды, характеризуется способом организации коммуникаций и строгостью требований по документированию. Модель должна определяться для каждого конкретного проекта и может меняться в зависимости от масштаба, новизны и критичности проекта, распределения участников, требований заказчика.

Предприятие как система требует системного подхода при реализации ИТ-проектов. Каждый проект должен рассматриваться в контексте и во взаимосвязи со смежными проектами. Чем сложнее проект, тем более формализованного подхода он требует. Именно формализованный и структурированный подход призван обеспечить реальную возможность выполнения поставленных задач. В современной практике модели разработки программного обеспечения многовариантны. Нет единственно верной для всех проектов, стартовых условий и моделей оплаты. Выбор зависит от специфики проекта, системы бюджетирования, субъективных предпочтений. Комплексный архитектурный подход (вариативное комбинирование формализованных и структурированных моделей) позволяет обеспечить реальную возможность выполнения информационных задач, позволяет сопоставить и привести в соответствие модель ведения бизнеса и используемые технологические подходы, сосредоточить ресурсы там, где они принесут максимальную отдачу.

Литература:

1. A Leader's Framework for Decision Making – Harvard Business Review <https://hbr.org/a-leaders-framework-for-decision-making>.
2. Barry Boehm, et al. (2000) Software cost estimation with COCOMO II. Englewood Cliffs, NJ: Prentice-Hall, 2000 www.systemsguild.com/riskology.
3. Bourque P., Fairley R. E. SWEBOK V 3.0. Guide to the Software Engineering Body of Knowledge. URL: <http://www.computer.org/portlet/swebok/SWEBOKv3.pdf>?
4. Chaikowska M. (2017) Methodological bases of IT-Project management with simulation modeling tools //Scientific Journal of Polonia University. PNAP. Periodyk naukowy Akademii Polonijnej, Częstochowa, Akademia Polonijna w Częstochowie, 2017, 21 (2017) nr 2, s. 148. – P. 55-66. – <https://nuife.org/index.php/pnap/article/view/135/134>.
5. Chaikowska M. (2016) Adoptive models of management modern IT-projects/ Materials I International Scientifical and Practical Conference Foresight-management: best world practice of development and integration of education, science and business. – Tbilisi: TSU, 2017. – p. 114-116.
6. Cobb G. (2016) Making Sense of Agile Project Management: Balancing Control and Agility, – N.Y.: Wiley, 2016. – 265 p.
7. Cockburn, A. (1999) Methodology per Project // Humans and Techno logy Technical Report HaT TR – 1999.
8. Cohn M. (2015) Succeeding with Agile: Software Development Using Scrum – Boston: Addison-Wesley Professional, 2015. – 504 p.
9. David Rubinstein (2017) Standish Group Report: There's Less Development Chaos Today. 2017 – <http://www.sdtimes.com/content/article.aspx?ArticleID=30247>.
10. ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes <https://www.iso.org/standard/43447.htm>.
11. Gartner Research (2017): <https://www.gartner.com/technology/>.

12. Kent Beck, and others, (2001) Manifesto for Agile Software Development, 2001 <http://www.agilemanifesto.org/>.
13. Microsoft Solutions Framework (2017), Microsoft Developer Network. URL: msdn.microsoft.com/ru-ru/library/jj161047.aspx.
14. M. Pomeroy-Huff, J. Mullaney, R. Cannon, M. Sebern, (2005) The Personal Software Process (PSP) Body of Knowledge, version 2.0, SPECIAL REPORT CMU/SEI, 2005.
15. Paulk, Mark C., and others, (1993) Capability Maturity Model for Software, Version 1.1 (CMU/SEI-93-TR-24). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1993.
16. PMBOK. (2014) Руководство к Своду знаний по управлению проектами, 5-е изд., PMI, 2014.
17. PRINCE2 Edition 2009: A Pocket Guide (Prince2: A Pocket Guide) by Bert Hedeman, Ron Seegers 1st (first) Edition (2009).
18. The Scrum Guide: <https://www.scrum.org/resources/scrum-guide>.
19. [XP] Jeffries, R., Beck, K., et al., (2015) Extreme Programming, as described on the web: <http://extremeprogramming.com>, <http://armaties.com/extreme.html>.
20. 1C: <https://www.softcom.ua/ru/1c/>.