

DOI: 10.15593/2499-9873/2019.2.02

УДК 519.714.2

С.С. Гусев

Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия

УПРАВЛЕНИЕ ПРОЦЕССОМ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Целью данной работы является обеспечение поддержки принятия решений по управлению процессом разработки ПО информационных систем. Новизна полученных результатов определяется тем, что разработанные модели и алгоритмы управления процессом разработки ПО информационных систем обеспечивают повышение качества управленческих решений за счет обеспечения учета неопределенности в оценках экспертов на начальном этапе проекта, автоматизированного расчета объема вносимых изменений, как доли требований, реализованных на предыдущих этапах/итерациях и затронутых при реализации новых требований, на основе вычисленных связей между элементами проекта и автоматизированного расчета оценок трудоемкости и сроков разработки ПО на основе систем нечеткого логического вывода, динамически подстраиваемых по мере накопления ретроспективных данных.

Ключевые слова: программное обеспечение, информационные системы, процесс управления, программные системы, автоматизированная система.

S.S. Gusev

V.A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences,
Moscow, Russian Federation

INFORMATION SYSTEMS SOFTWARE DEVELOPMENT PROCESS CONTROL

The purpose of this work is to provide decision support for the management of software development information systems. The novelty of the results is determined by the fact that the developed models and algorithms for managing the process of developing information systems software provide an increase in the quality of management decisions by ensuring the accounting of uncertainty in the estimates of experts at the initial stage of the project, automated calculation of the volume of changes as a proportion of requirements implemented in the previous stages/iterations and affected by the implementation of new requirements, based on the calculated relationships between the elements of the project and the automated calculation of estimates of the complexity and timing of software development based on fuzzy inference systems, dynamically adjusted as retrospective data are accumulated.

Keywords: software, information systems, control process, software systems, automated system.

Введение

Стратегия развития информационного общества в РФ на 2017–2030 годы¹ акцентируется на вопросах развития единого информационного пространства, формирования национальной цифровой экономики, обеспечения национальных интересов и реализации стратегических национальных приоритетов и, в частности, на развитии критической информационной инфраструктуры РФ. Все это обуславливает высокую потребность в разработке информационных систем различного назначения.

Для реализации целей и задач информационных систем и обеспечения функционирования как отдельных, так и комплекса технических средств необходимо программное обеспечение (ПО) – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ.

Процесс разработки ПО представляет собой комплекс мероприятий по сбору и разработке требований и преобразованию их в ПО, осуществляемый под управлением представителей заказчика ПО. Управленческие решения считаются эффективными, если разработанное ПО работоспособно при выполнении ограничений, заданных на этапе планирования. Сложность принятия управленческих решений при разработке ПО обуславливается существенным значением неформализуемой категории «творчество», а также необходимостью разрешения противоречий при определении сроков разработки (которые минимизируются), списка требований к функциональности ПО (которые максимизируются) и доступным ресурсам и бюджету (которые минимизируются).

С учетом того, что заказчик разработки ПО, как правило, одновременно сопровождает несколько десятков проектов по разработке ПО (по разработке нового ПО и модернизации эксплуатируемого ПО), наличие названных противоречий неизбежно приводит к решениям, обуславливающим наличие неуспешно завершенных проектов. Как показывает практика, наиболее часто встречающимися причинами неуспешного завершения проектов по разработке ПО являются:

- ошибки выбора модели процесса разработки ПО;
- ошибки оценивания сроков и трудоемкости проекта на начальных стадиях проекта (формирование недостижимых целей);

¹ Указ Президента РФ от 09.05.2017 г. № 203.

– ошибки в оценке сроков и трудоемкости работ в ходе выполнения проекта, приводящие к невозможности корректировки плана проекта или изменения ресурсов в случае возникновения отклонений и, как следствие, нарушению проектных ограничений;

– ошибки определения потребного объема регрессионного тестирования в итерационных моделях разработки ПО, что, как следствие, приводит или к увеличению срока разработки (при полном тестировании по всем тестовым сценариям), или к увеличению числа пропущенных дефектов при сильно уменьшенном объеме тестирования.

Вопросы поддержки принятия управленческих решений при разработке ПО исследуются и разрабатываются с 1960-х гг. Американскими учеными в интересах управления процессом разработки ПО в Министерстве обороны США созданы известные методы (PERT, функциональных точек, СОСОМО), обеспечивающие получение реалистичных оценок сроков разработки ПО в условиях ограничений по срокам разработки, размерности множества требований к функциональности ПО, доступным ресурсам и бюджету. Аналогичные решения и разработки отечественных ученых, уточняющие отдельные аспекты названных методов, появились только в 1990-х гг.

В настоящее время методы и технологии поддержки принятия управленческих решений при разработке ПО получили новый импульс развития, обусловленный появлением гибких методологий разработки ПО, работы таких ученых, как Т. Демарко, Т. Листер, К. Вигерс, А. Кокберн, С. Макконел, Г.Н. Калянов, В.В. Липаев, А.М. Вендров, А.А. Демирский, Ю.М. Мадорская и др.

Однако модели и алгоритмы управления процессом разработки ПО, учитывающие специфику разработки ПО для информационных систем, не разработаны, что обусловило актуальность исследования.

1. Постановка задачи

1.1. Определение процесса разработки ПО информационных систем назначения

Процесс разработки программного обеспечения представляет собой комплекс мероприятий по сбору и разработке требований и преобразованию их в программное обеспечение. В работе под понятием «требование» будет использоваться общее определение требования,

данное в своде знаний по программной инженерии третьей версии [1], как свойство, представленное чем-либо, для решения некоторой проблемы реального мира. Более детально, в зависимости от методологии проектирования и обработки требований в организации, возможно деление требований на различные типы (как пример [1–4]: функциональные требования, нефункциональные требования, атрибуты качества, ограничения, бизнес-требования и бизнес-правила).

Процесс разработки ПО, как правило, протекает в условиях трех основных видов ограничений: ограничения по срокам, ограничения по составу работ, ограничения по имеющимся ресурсам и бюджету. В этом случае понятие процесса разработки ПО можно считать синонимом термина «проект разработки ПО», так как, в соответствии с ГОСТ Р 54869–2011², проект – это комплекс взаимосвязанных мероприятий, направленный на создание уникального продукта или услуги в условиях временных и ресурсных ограничений. Некоторые модели разработки ПО допускают возможность разработки ПО без ограничения по времени, так называемые гибкие методологии, но в этом случае временные ограничения накладываются на этапы выполнения работ, в то время как самих этапов может быть большое число. В связи с этим понятие процесса будем считать более общим, но на практике в большинстве случаев, требуется получение работающего ПО в заданный промежуток времени, поэтому в дальнейшем будем использовать оба термина как синонимы.

Согласно ГОСТ РВ 51987, информационная система – это автоматизированная система, результатом функционирования которой является представление выходной информации для последующего использования. Особенности информационных систем, обуславливающих сложность их разработки, являются [5]:

- длительный срок хранения данных (десятилетия), данные при этом способны пережить несколько поколений информационных систем, предназначенных для их обработки, аппаратных средств, операционных систем и компиляторов;

- структура данных претерпевает изменения в процессе развития системы и изменения предметной области в целях сохранения но-

² ГОСТ Р 54869–2011. Проектный менеджмент. Требования к управлению проектом. М.: Стандартинформ, 2012. 11 с.

вых объемов данных без воздействия на старые, уже хранящиеся в системе;

- большие объемы данных;
- десятки и сотни пользователей, одновременно модифицирующих данные в информационной системе;
- интеграция с большим количеством других информационных систем, разработанных в разное время с использованием разных технологий.

Определение управления проектом дано в ГОСТ Р 54869–2011 как планирование, организация и контроль трудовых, финансовых и материально-технических ресурсов проекта, направленные на эффективное достижение целей проекта. Под эффективностью понимается соблюдение проектных ограничений, заданных на этапе планирования проекта. На основании этого определения управленческое решение лица, принимающего решение, будет считаться качественным, если оно способствует достижению целей проекта, в области разработки ПО – выпустить работоспособное программное обеспечение без нарушения заданных на этапе планирования ограничений.

1.2. Современное состояние отрасли по разработке ПО

На основе ежегодного аналитического отчета российского некоммерческого партнерства РУССОФТ [6], объединяющего в себе разработчиков программного обеспечения, в России по состоянию на 2013 г. зарегистрировано более 1400 компаний, занимающихся разработкой программного обеспечения и общий объем экспорта программной продукции составил 4,6 млрд долл. При этом далеко не все проекты завершаются успешно. Согласно исследованию Standish Group, в период с 2003 по 2015 г., успешно завершилось только 6 % проектов по разработке программного обеспечения, 42 % было завершено неудачно и 52 % проектов предоставили недостаточно данных, или имеют выход за рамки бюджета расписания, или разработанный продукт не соответствовал требованиям заказчика.

Согласно совместному исследованию специалистов консалтингового агентства McKinsey & Company и Оксфордского университета, после обследования 5400 проектов по разработке программного обеспечения, начальный бюджет которых составил более 15 млн долл., установлено, что 17 % проектов было на грани срыва, в 45 % проектов

был превышен бюджет, 7 % проектов завершилось позже запланированного, и в 56 % проектах содержание проекта не соответствовало начальному. По статистике за 2000 по 2019 г., собранной на основе анализа различных источников [7–9], согласно диаграмме, представленной на рис. 1, уровень успешности завершения программных проектов остается примерно постоянным и составляет порядка 30 %.

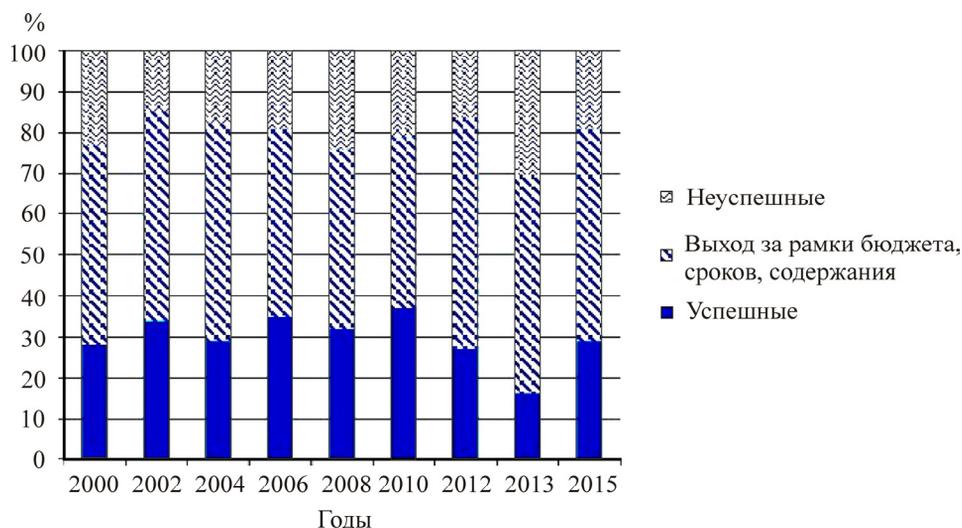


Рис. 1. Успешные и неуспешные проекты по разработке программного обеспечения за период 2000–2015 гг.

Стратегия развития информационного общества в РФ на 2017–2030 гг. акцентируется на вопросах формирования национальной цифровой экономики, обеспечении национальных интересов и реализации стратегических национальных приоритетов, и в частности – на развитии критической информационной инфраструктуры в сфере обороны Российской Федерации. Все это обуславливает высокую потребность в разработке информационных систем военного назначения.

На основании опроса экспертов в области разработки программного обеспечения выделены основные группы факторов, влияющих на процесс разработки ПО. Установлено следующее распределение ответов экспертов: проекты срываются только по техническим причинам – 1 %, по причине только организационных сложностей – 19 %, в равной степени и организационных и технических – 21 %, и в большей степени организационных, чем технических – 59 %.

В соответствии со стандартами по управлению проектами ISO 21500:2012 и PMBOK v6, оценку сроков выполнения проектов можно представить в виде следующих процессов: планирование содержания проекта и на основе полученного содержания планирование сроков выполнения проекта. Модель взаимодействия этих процессов в нотации IDEF0 представлена на рис. 2.

Декомпозиция процесса планирования содержания выделяет следующие подпроцессы:

- сбор требований,
- определение содержания,
- формирование иерархической структуры работ.

Диаграмма декомпозиции процесса планирования содержания проекта представлена на рис. 2. В результате выполнения процесса сбора требований формируется документация с описанием требований и матрица отслеживания требований. На основе этой информации формируется высокоуровневое описание содержания проекта, после чего – иерархическая структура работ, с выделением пакетов работ и отдельных задач на разработку.

1.3. Ретроспективный анализ моделей процесса разработки ПО

Основные процессы разработки ПО схожи с процессами разработки в других инженерных областях и состоят из следующих стадий: проектирование, разработка (создание образца изделия), испытания (тестирование), серийное производство, сопровождение. Процесс, начинающийся с момента принятия решения о разработке программного обеспечения и завершающийся его изъятием из эксплуатации, носит название жизненного цикла программного обеспечения. За время развития индустрии разработки программного обучения модели жизненного цикла также претерпели значительные изменения, в частности появилось большое количество «легковесных», или agile, методологий, с возможностью «легкого» внесения изменений в структуру разрабатываемых программ.

Одной из первых моделей разработки программного обеспечения является водопадная (waterfall), или каскадная, модель жизненного цикла. Он также носит название классического жизненного цикла разработки ПО, отчасти в связи с тем, что эта модель повторяет последовательность действий при разработке инженерных решений в других

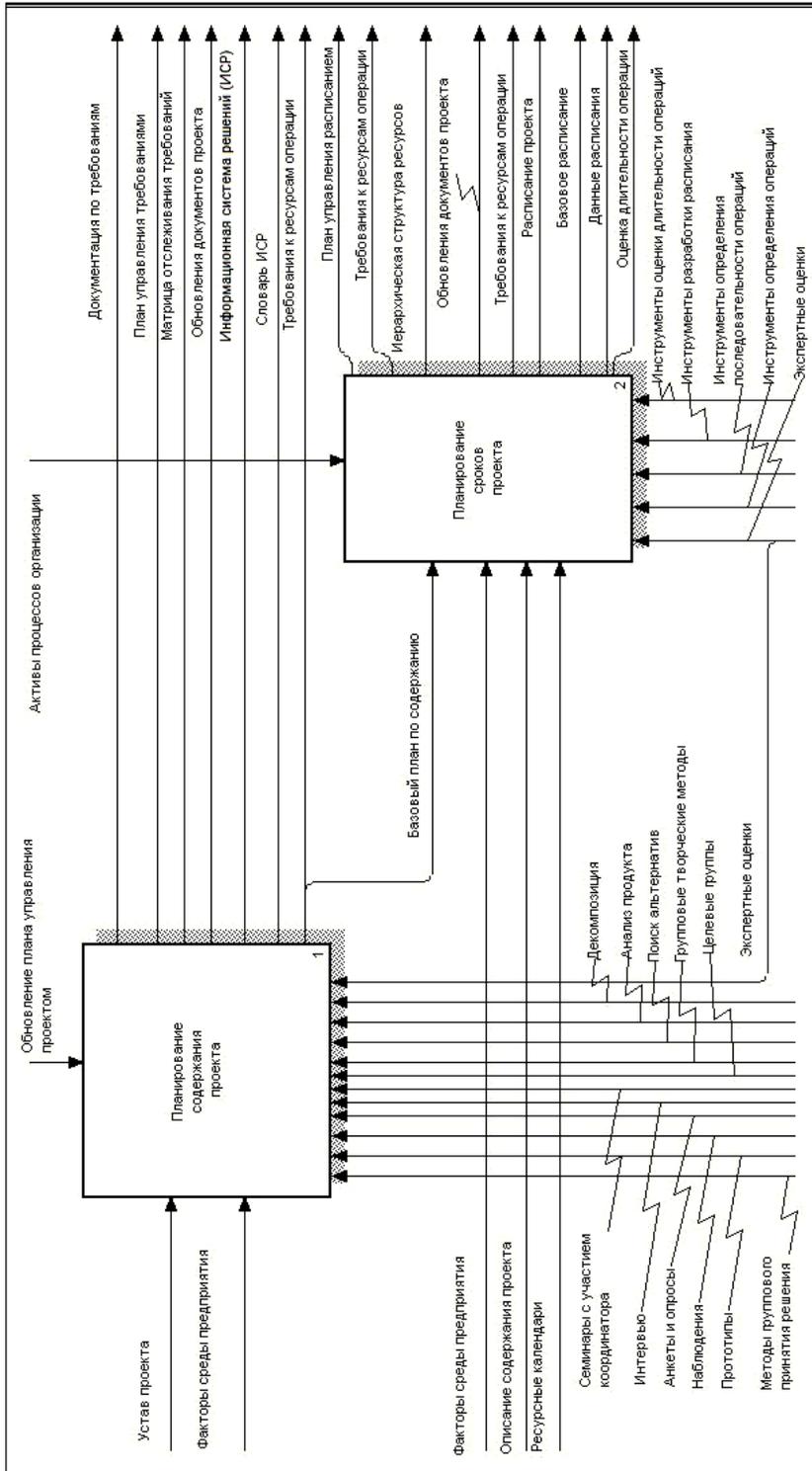


Рис. 2. Диаграмма верхнего уровня планирования сроков и содержания проекта

областях науки и техники. Впервые данный подход как адаптированный для разработки ИТ-продуктов упомянул Герберт Бенингтон (Herbert D. Benington) в своей презентации 29 июня 1956 г. на Симпозиуме по передовым методикам для цифровых компьютеров (Symposium on advanced programming methods for digital computers). Но более формально этот жизненный цикл был описан Уинстоном Ройсом в статье 1970 г. [10]. Итак, он считается классическим и используется в качестве основы для сравнения с другими моделями жизненного цикла ПО.

По классификации, представленной в своде знаний по программной инженерии SWEBOOK, третья версия которого вышла в 2013 г., модели жизненного цикла разделяются на две категории – линейные и итерационные модели. Исторически первой в системе разработки была водопадная модель, но в связи с тем, что в отсутствие полноты требований возникала необходимость в их уточнении и внесении дополнительных изменений в проект на стадии реализации или тестирования, были предложены различные итеративные модели, которые, по сути, отличаются только количеством итераций, их размером и действиями, выполняемыми при подготовке и завершении каждой итерации. В связи с этим был выделен класс гибких методологий разработки, в которых размер итераций был сравнительно маленьким (занимал порядка недели, а в некоторых – до одного дня). Подобные подходы применяются в основном при разработке программного обеспечения в условиях изменяющихся требований. В условиях высокой неопределенности предметной области применяются итеративные подходы, которые относятся к семейству *Agile*, наиболее известны из них: экстремальное программирование (XP), *Scrum*, *Kanban*, *OpenUP*, *FDD*. Основной идеей, лежащей в основе этих процессов, является частая смена итераций с получением на каждой итерации работающего программного образца, пусть и реализующего не весь функционал, заложенный на этапе постановки задачи и анализа требований.

Главным достоинством этого подхода является возможность быстрого реагирования на возникающие изменения в процессе разработки. Но перед внедрением процессов семейства *Agile* стоит ряд препятствий:

– в промышленной разработке требуется полный комплект документации на разработанное программное обеспечение, что является

ся противоречием основному принципу гибких процессов разработки: «работающий программный продукт важнее полной документации»;

– снижение качества получаемого программного продукта в тех случаях, когда невозможно получить новую версию за короткую итерацию;

– проекты по промышленной разработке ПО, как правило, выполняются в жесткой кадровой структуре организации, что препятствует внедрению принципа использования самоорганизующихся команд разработчиков.

2. Формирование структуры системы поддержки принятия решения

2.1. Структура системы поддержки принятия решения

При разработке системы поддержки принятия решений на практике была предложена следующая структура системы (рис. 3).

В работе в качестве системы проектирования и управления требованиями взята система Enterprise Architect, при этом все данные моделей и требования хранятся на сервере в отдельной базе данных, с целью обеспечения совместной работы нескольких аналитиков и разработчиков. Поскольку все данные хранятся в статическом виде, необходим отдельный компонент системы, представленный в виде отдельного приложения, которое отслеживает изменения в моделях и требованиях. Найденные изменения передаются в подсистему оценки объема изменений, связанную с системами управления задачами и системой контроля версий программного кода.

В качестве системы управления задачами был выбран продукт Jira от Atlassian, в качестве системы контроля версий была выбрана система Mercurial. В рамках предложенной архитектуры и решаемых задач системы управления задачами должны удовлетворять следующим требованиям:

– должна существовать интеграция между системой контроля версий и системой управления задачами;

– должна существовать интеграция между системой проектирования (управления требованиями) и системой управления задачами;

– у системы управления задачами должен быть интерфейс для подключения внешних систем, позволяющий управлять задачами (по-

лучать список задач, получать параметры отдельных задач, создавать новые задачи).

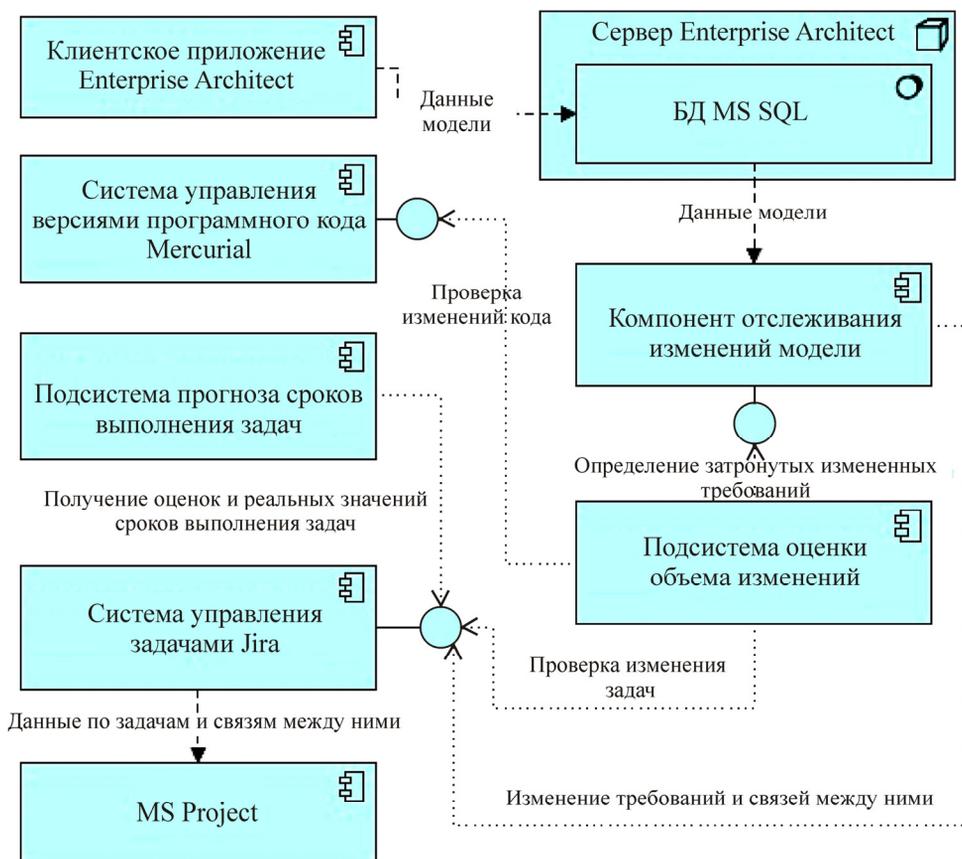


Рис. 3. Структура системы поддержки принятия решений при управлении проектом в нотации Archimate

В интерфейсе для подключения внешних систем предусмотрены связи с подсистемами прогноза сроков выполнения задач, оценки объема изменений и компонентом отслеживания изменений модели.

Для построения сетевого графика и оценки сроков выполнения задач на основе скорректированной трудоемкости использовалась система MS Project.

2.2. Проверка эффективности предложенных решений в реальных проектах

Проверка предложенных решений проводилась на пяти проектах по разработке ПО:

- **проект 1:** экспертная система верификации биотерактов (БТ) в Вооруженных Силах Российской Федерации [11];
- **проект 2:** распределенный банк данных по техническим средствам медицинской службы Вооруженных Сил Российской Федерации;
- **проект 3:** медицинский регистр военнослужащих, получивших поражения в бою;
- **проект 4:** платформа по проведению групповой экспертизы и оцениванию технического уровня сложных систем;
- **проект 5:** система обеспечения сетевого взаимодействия участников распределенной редакционной коллегии в процессах создания, редактирования, рецензирования и публикации материалов электронных изданий.

Методика проведения оценки заключалась в следующем:

1. На начальном этапе проводилась предварительная оценка с использованием предложенной модели оценки трудоемкости и сроков. Фиксировалась максимальная и минимальная оценка срока и трудоемкости разработки.

2. В процессе разработки ПО, перед началом каждого инкремента формируется прогнозная оценка срока и трудоемкости выполнения инкремента без применения моделей и алгоритмов поддержки принятия решения (ППР) и с использованием алгоритмов поддержки принятия решения.

3. Модели поддержки принятия решения корректируются в ходе выполнения проекта на каждой итерации.

4. Перед началом регрессионного тестирования проводится отбор требований для повторного тестирования из числа реализованных на предыдущих этапах с использованием алгоритма автоматизированного выявления связей между элементами проекта на основе данных из систем хранения исходного кода. Для оценки качества работы алгоритма экспертом проводилась аналогичная работа по отбору требований вручную, без применения алгоритма. По завершении проекта производится оценка точности прогноза на ранней стадии проекта и сравнение

результатов экспертных оценок с применением и без применения ППР на каждом инкременте проекта.

Общие характеристики проектов и результаты прогноза на начальной стадии проекта приведены в табл. 1.

Таблица 1

Общие характеристики проектов

Показатель	Проект				
	1	2	3	4	5
Размер команды в среднем, чел.	32	31	29	20	19
Минимальная прогнозируемая длительность по модели, дни	320	480	210	180	190
Максимальная прогнозируемая длительность по модели, дни	400	620	340	250	220
Минимальная прогнозируемая трудоемкость по модели, ч/дни	11 650	14 800	5240	4200	3900
Максимальная прогнозируемая трудоемкость по модели, ч/дни	16 600	22 800	11 200	6900	5800
Фактическая длительность, дни	393	587	259	232	321
Фактическая трудоемкость, ч/дни	12 611	18 502	7562	4830	6208
Число инкрементов	4	6	3	3	3
Общее число итераций	34	54	19	21	23

Из пяти рассмотренных проектов четыре уложились в рассчитанный интервал оценки. Сводная информация по проекту 1 приведена в табл. 2.

Таблица 2

Сводная информация по проекту 1

Показатель	Инкремент			
	1	2	3	4
Прогнозируемый срок инкремента без ППР	60	90	95	90
Прогнозируемая трудоемкость инкремента без ППР	1800	2800	3400	2800
Прогнозируемый срок инкремента с ППР	79	103	125	105
Прогнозируемая трудоемкость инкремента с ППР	2230	3340	3720	2920
Фактическая длительность инкремента	83	102	122	86
Фактическая трудоемкость инкремента	2674	3267	3914	2756

Окончание табл. 2

Показатель	Инкремент			
	1	2	3	4
Относительная ошибка по срокам без ППР	23	12	27	4
Относительная ошибка по трудоемкости без ППР	874	467	514	44
Относительная ошибка по срокам с ППР	4	1	3	19
Относительная ошибка по трудоемкости с ППР	444	73	194	164
Доля измененных требований, выявленных с применением алгоритма выявления связей между элементами проекта разработки ПО, %	–	0,733 33	0,8	0,448 7
Доля измененных требований на основе экспертной оценки, %	–	0,8	0,742 86	0,371 8
Ошибка алгоритма выявления связей между элементами проекта разработки ПО	–	0,066 67	0,057 14	0,076 9

3. Результат расчетов зависимости срока проекта от возможности завершения проекта к заданному сроку

Согласно зависимости, представленной на рис. 4, завершить выполнение проекта со степенью возможности 0,2 можно за 186 рабочих дней, со степенью возможности 0,5 – за 227, со степенью возможности 0,8 – за 281 день.

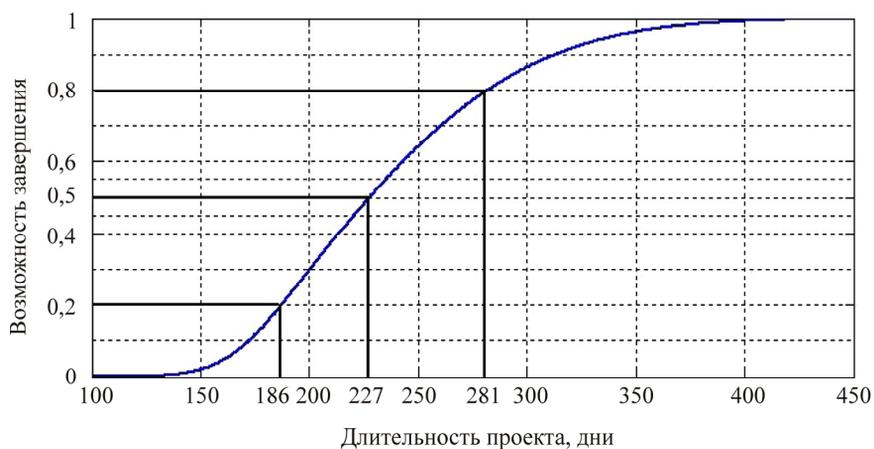


Рис. 4. Рассчитанная зависимость срока проекта и возможности завершения проекта к заданному сроку

Заключение

Наличие противоречия между необходимостью обоснования сроков и трудоемкости разработки ПО при формировании технического задания на его разработку, осуществления непрерывного контроля реализации заданных требований и отсутствие необходимых компетенций у заказчика разрабатываемого ПО обуславливает задачу разработки моделей и алгоритмов поддержки принятия решений по управлению процессом разработки ПО.

Применение результатов исследования при разработке ПО пяти информационных систем обеспечило информационную поддержку принятия решения на этапах предварительной оценки сроков и трудоемкости разработки ПО, выбора модели жизненного цикла ПО, динамической оценки сроков и трудоемкости работ в ходе выполнения проекта, определения объема регрессионного тестирования.

Список литературы

1. Guide to the Software Engineering Body of Knowledge Version 3.0. – Washington D.C.: IEEE Computer Society, 2013. – 355 p.
2. Davis A. Just Enough Requirements Management: Where Software Development Meets Marketing. – New York: Dorset House, 2005. – 240 p.
3. Вигерс К., Битти Д. Разработка требований к программному обеспечению. – М.: Русская редакция, 2014. – 736 с.
4. Гольфанд И.Я., Хлебутин П.С. Оценка трудозатрат разработки программной компоненты // Труды ИСА РАН. – 2005. – Т. 15. – С. 125–135.
5. Фаулер М. Архитектура корпоративных программных приложений. – 2-е изд. – М.: Вильямс, 2006. – 544 с.
6. Десятое ежегодное исследование Российской индустрии экспортной разработки программного обеспечения / Руссофт. – М., 2013. – 183 с.
7. Lars Mieritz Gartner Survey Shows Why Projects Fail, 2012. – URL: <http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-rojects-fail/> (accessed 02 May 2019).
8. Rupinder K., Sengupta J. Software Process Models and Analysis on Failure of Software Development Projects // International Journal of Scientific & Engineering Research. – 2011. – Vol. 2, iss. 2. – P. 1–4.
9. Stanish Group Chaos report 2014. – URL: <http://www.projectsart.co.uk/docs/chaos-report.pdf> (accessed 02 May 2019).
10. Royce W. Managing the Development of Large Software Systems // TRW. – 1970. – August. – P. 328–338.

11. Экспертно-аналитическое обоснование приоритетных направлений совершенствования системы предупреждения биологических террористических актов / А.В. Богомолов, Т.В. Зуева, С.С. Чикова, М.С. Голосовский // Информатика и системы управления. – 2009. – № 4 (22). – С. 134–136.

References

1. Guide to the Software Engineering Body of Knowledge, Version 3.0. Washington D.C., IEEE Computer Society, 2013. 355 p.

2. Davis A. Just Enough Requirements Management: Where Software Development Meets Marketing. New York, Dorset House, 2005. 240 p.

3. Vigers K., Bitti D. Razrabotka trebovanij k programmnomu obespecheniyu [Software requirements development]. Moscow: Russkaya redakciya, 2014. 736 p.

4. Gol'fand I.Ya., Hlebutin P.S. Ocenka trudozatrata razrabotki programmnoj komponenty [Evaluation of labor costs of software component development] Trudy ISA RAN. 2005. Vol. 15. pp. 125-135.

5. Fauler M. Arhitektura korporativnyh programmnyh prilozhenij [Enterprise software application architecture]. Moscow, Vil'yams, 2006. 544 p.

6. Desyatoe ezhegodnoe issledovanie Rossijskoj industrii eksportnoj razrabotki programmogo obespecheniya [Tenth annual survey of the Russian export software development industry]. Moscow, Russoft. 2013. 183 p.

7. Lars Mieritz Gartner Survey Shows Why Projects Fail, 2012, available at: <http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-projects-fail/>.

8. Rupinder K., Sengupta J. Software Process Models and Analysis on Failure of Software Development Projects. International Journal of Scientific & Engineering Research. 2011, iss. 2, vol. 2, pp. 1-4.

9. Stanish Group Chaos report 2014. available at: <http://www.projectsmart.co.uk/docs/chaos-report.pdf> (accessed 02.05.2019).

10. Royce W. Managing the Development of Large Software Systems TRW. 1970, pp. 328-338.

11. Bogomolov A.V., Zueva T.V., CHikova S.S., Golosovskij M.S. Ekspertno-analiticheskoe obosnovanie prioritetnyh napravlenij sovershenstvovaniya sistemy preduprezhdeniya biologicheskikh terroristicheskikh aktov [Expert-analytical substantiation of priority directions of improving the system of prevention of biological terrorist acts]. Informatika i sistemy upravleniya. 2009, iss. 4 (22), pp. 134-136.

Получено 02.05.2019

Сведения об авторе

Гусев Сергей Сергеевич (Москва, Россия) – соискатель, Институт проблем управления им. В.А. Трапезникова РАН (117997, г. Москва, ул. Профсоюзная, 65, e-mail: gs-serg@mail.ru).

About the author

Sergei S. Gusev (Moscow, Russian Federation) – Ph.D. Student, V.A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences (65, Profsouznaya st., Moscow, 117997, Russian Federation, e-mail: gs-serg@mail.ru).