

А. Е. Колоденкова

## АКТУАЛЬНЫЕ ПРОБЛЕМЫ СОВРЕМЕННОЙ ПРОГРАММНОЙ ИНЖЕНЕРИИ: КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ И ЖИЗНЕСПОСОБНОСТЬ ПРОГРАММНОГО ПРОЕКТА

Обсуждаются проблемы концептуального проектирования программного обеспечения и анализа жизнеспособности программных проектов. Обобщены и систематизированы различные научные взгляды как отечественных, так и зарубежных ученых-специалистов на проблему анализа жизнеспособности программных проектов. Предложен вероятностно-статистический подход к оценке жизнеспособности проектов, основанный на сравнении альтернатив реализации проекта по критерию вероятного выполнения совокупности работ за выделенное на проект время. Отмечено, что для оценки жизнеспособности проектов наиболее перспективными являются комплексные подходы, получившие название высоких статистических технологий. *Концептуальное проектирование программного обеспечения; проблемы анализа жизнеспособности проекта; виды, критерии и показатели жизнеспособности проекта; задачи и методы оценки жизнеспособности проекта; вероятностно-статистический подход к оценке жизнеспособности проекта*

### ВВЕДЕНИЕ

Среди пяти ключевых направлений модернизации российской экономики выделяются стратегические информационные технологии, базирующиеся на создании суперкомпьютеров и разработке программного обеспечения (ПО) [1].

Современное ПО относится к классу больших программных проектов (ПП), под которыми понимается разновидность инновационного проекта, рассматриваемая как интеллектуальная деятельность коллектива разработчиков, направленная на достижение заранее определенного результата / цели при заданных ресурсных и временных ограничениях с учетом требований к качеству. При этом результатом ПП является создание уникального программного продукта, т. е. ПО в виде совокупности программ обработки данных и необходимых для их эксплуатации документов [2–4].

Отличительной особенностью ПП является его разработка и реализация в условиях риска, возникающего в силу неопределенности факторов внутренней и внешней среды проекта, которые могут привести либо к ухудшению качественных показателей разрабатываемого программного продукта, либо к превышению бюджетного ассигнования и / или нарушению сроков осуществления проекта, либо просто к его провалу. В связи с этим все большую актуаль-

ность и значимость с финансовой, экономической и других точек зрения [5, 6] приобретают проблемы концептуального проектирования ПО и, в частности, проблема анализа жизнеспособности, т. е. осуществимости ПП.

Несмотря на большое число работ, посвященных различным научно-исследовательским и опытно-конструкторским проектам (например [7, 8]), проблема концептуального проектирования ПО до сих пор остается открытой. В настоящей работе рассмотрены некоторые аспекты данной проблемы и акцентировано внимание на актуальных вопросах анализа жизнеспособности ПП.

### 1. «КРИЗИС ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Объективная потребность контролировать процесс разработки ПО, прогнозировать и гарантировать стоимость, а также сроки и качество его выполнения привела к появлению так называемой программной инженерии (software engineering) [9] – области компьютерных наук и технологий, связанной с индустриальным производством высококачественных, экономичных и надежных программных продуктов для решения определенной проблемы в виде комплекса взаимосвязанных программ и сопроводительной технической документации по его установке, настройке и использованию.

В современной государственной политике важную роль играют так называемые инновационные проекты, которые реализуются в виде больших, сложных, крупномасштабных межотраслевых проектов по созданию, освоению и распространению технологий, способствующих

Контактная информация: (347) 272-74-65

Автор выражает благодарность зам. главного редактора журнала «Информационные технологии» д-ру техн. наук, проф. Н. Б. Филимонову за высказанные замечания и пожелания по улучшению статьи.

щих кардинальным изменениям в технологическом базисе экономики, а также по развитию фундаментальных исследований, научно-техническому обеспечению социальных программ и международного сотрудничества.

Несмотря на определенные успехи, достигнутые программной инженерией на современном этапе развития, реализация подавляющего большинства крупных ПП не укладывается в запланированные сроки, характеризуется низкой производительностью, существенным отставанием от графика, а созданные в результате программные продукты часто не вполне отвечают требованиям заказчиков-пользователей. Более того, все чаще ПП оказываются просто проваленными, т. е. имеет место процесс утраты их жизнеспособности. Так, согласно статистике «Хаос» о состоянии дел в программной индустрии в 2008 году, опубликованной компанией Standish Group [10], из 30 тыс. программных проектов 32% проектов завершились успешно, 44% проектов завершились с проблемами (превысили бюджет, сроки и пр.) и 24% проектов полностью провалились. Все это является проявлением «кризиса ПО», в условиях которого, по мнению Э. Йордана, «Безнадежные проекты являются нормой, а не исключением».

В качестве примера можно привести ПО компании Primavera Systems, Inc. (SureTrak и Primavera Project Planner Professional), предназначенное для создания автоматизированных систем управления проектами с настройкой на решение любой проектно-ориентированной задачи в любой сфере деятельности. Можно указать также на ПО компании Microsoft (Microsoft Office Project Professional, Microsoft Office Project Server, Microsoft Office Project Portfolio Server), позволяющее управлять проектными работами, планами, финансами и т. п.

## 2. КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В связи с «кризисом ПО» в программной инженерии [9] все более важная роль отводится анализу жизнеспособности ПП – стадии жизненного цикла проекта, позволяющего оценивать возможность его осуществления (сохранение жизнеспособности), формулировать его общественную и коммерческую значимость, а также выявлять ключевые факторы успеха его реализации. В настоящей работе под жизнеспособностью проекта понимается наличие необходимых ресурсов, а также условий для его реализации.

Из-за сложности процесса реализации ПП дать универсальный подход к его разделению на стадии весьма затруднительно. В связи с этим В. В. Липаев подчеркивает, что существующие стандарты ГОСТ ЕСПД, предусматривающие стадии и этапы (создание, сопровождение и совершенствование) жизненного цикла ПП, «...отражены недостаточно, а многие их положения устарели». Решая данную задачу, разработчики ПО должны руководствоваться своей ролью в проекте, накопленным опытом и конкретными условиями его выполнения. Здесь можно согласиться с мнением И. И. Мазура: «Универсального подхода к разделению процесса реализации проекта на фазы не существует». В настоящей работе за основу принята структурная схема жизненного цикла ПП, представленная на рис. 1 [6].

Здесь весь жизненный цикл проекта разделяется на две стадии: концептуальное и материальное проектирование ПО.

Концептуальное проектирование – это процесс концептуальной разработки ПО, который включает два этапа: формулирование проекта (анализ идеи, выбор целей и постановка задач) и анализ жизнеспособности проекта (сбор, анализ и структурирование информации о проекте в форме, позволяющей принимать решение о возможности его осуществимости и способах реализуемости).

Материальное (рабочее или техническое) проектирование – это процесс физической разработки ПО, который включает три этапа: разработка проекта (происходит детальное планирование работы над проектом «от и до»), выполнение проекта (происходит непосредственная работа по реализации проекта) и завершение проекта (происходит полная реализация задуманного проекта, проводится анализ результатов).

Этап анализа жизнеспособности проекта (именуемый часто этапом принятия решений по жизнеспособности проекта, принятия управленческих решений) – это процесс анализа, прогнозирования и оценки ситуации, выбора и согласования наилучшего альтернативного варианта достижения поставленной цели.

Следует отметить, что понятие «наилучшая альтернатива» субъективно. Однако субъективизм проявляется на всех этапах принятия решения. Рациональная технология принятия решений призвана, прежде всего, минимизировать возможные ошибки с помощью применения специально разработанных методов для достижения наилучшего результата. С инфор-

мационной точки зрения в процессе принятия решений происходит уменьшение неопределенности.

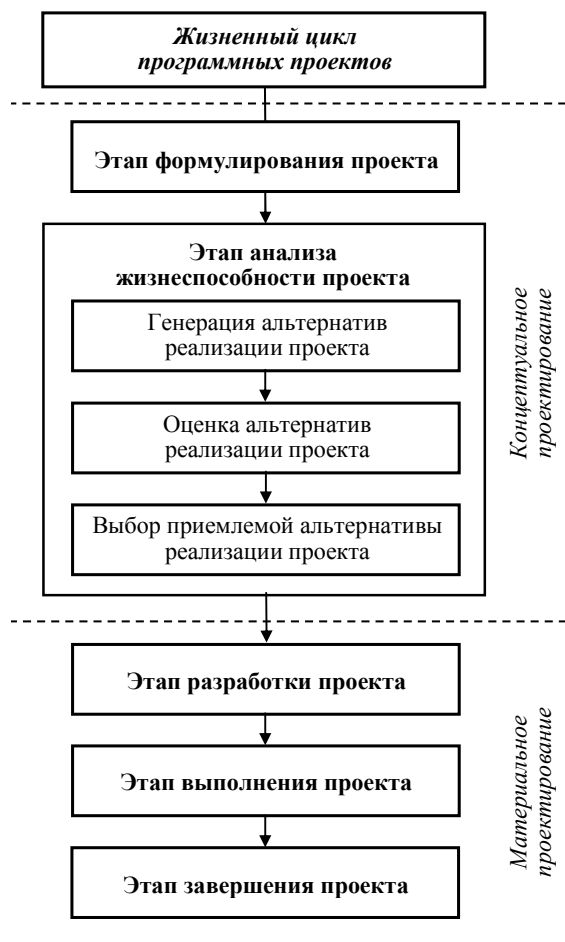


Рис. 1. Структура жизненного цикла ПП

Типовые процессы принятия решений, реализуемые в самых различных областях деятельности, имеют много общего, поэтому необходимо разработать некоторую схему процесса принятия решения, устанавливающую наиболее целесообразный набор и последовательность действий. Данную схему не следует рассматривать как жесткий алгоритм принятия управленческого решения.

В настоящее время разработано много различных подходов к выделению этапов принятия решений.

Например, А. М. Чуйкин предложил [11] этапы подготовки решений, включающие следующие шаги: построение взаимоотношений в организации для решения проблемы, формирование команды для подготовки решения, диагностика ситуации, разработка и обоснование системы целей; определение и анализ проблемы, формулировка критериев и ограничений,

выдвижение и анализ альтернатив, оценка альтернатив и последствий, выбор альтернативы. Мескон (M. Maskon), Альберт (M. Albert), Хедоури (F. Hedouri) и др. сокращают этот перечень, включая следующие «этапы рационального решения проблем организации»: диагностика проблемы, формулировка ограничений и критериев принятия решения, определение альтернатив, оценка альтернатив, выбор альтернативы, реализация и обратная связь [12].

Таким образом, проведенный анализ подходов к выделению этапов принятия решений позволяет сделать вывод о том, что предложенные в литературе этапы различаются лишь в деталях, но в целом дают верное описание логики развития этого процесса. Далее автором предлагаются следующие основные этапы анализа жизнеспособности проекта:

- генерация альтернатив реализации проекта (разработка возможных способов достижения поставленной цели);
- оценка альтернатив реализации проекта (расчет показателей, характеризующих достоинства и недостатки альтернатив);
- выбор приемлемой альтернативы реализации проекта (выбор альтернативы с наиболее благоприятными последствиями).

### 3. ПРОБЛЕМНАЯ ТРИАДА КОНЦЕПТУАЛЬНОГО ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В общем случае успех разработки больших ПП во многом определяется триадой проблем концептуального проектирования ПО, представленной на рис. 2 [13].

Успешность выполнения ПП (по стоимости, срокам и качеству разработки), возможные его риски можно спрогнозировать на начальных стадиях жизненного цикла проекта, однако такой прогноз существенно усложняется наличием следующих проблем.

Проблемы организации деятельности команды разработчиков ПП. Одним из главных препятствий успешной реализации больших ПП является нехватка или высокая занятость разработчиков (исполнителей, IT-разработчиков) проекта требуемой квалификации. В связи с этим уже на стадии концептуального проектирования ПО необходимо четко структурировать работы, определить обязанности разработчиков, их квалификацию и численность, что позволит заказчику оценить реальную потребность в человеческих ресурсах и снизить риск затягивания сроков выполнения проекта.



Рис. 2. Проблемная триада концептуального проектирования ПО

Действительно, согласно мнению гендиректора компании Primavera Systems, Inc. Коппельмана (J. Korrelman), «Ключевое условие успешной реализации проекта – тщательный подбор людей, обладающих требуемой квалификацией» на основе принципа «синхронизации проектов и человеческих усилий».

Накопленный к настоящему времени опыт реализации ПП показывает, что это сложная и трудоемкая работа, требующая высокой квалификации участвующих в ней специалистов. Формирование эффективной, высококвалифицированной команды является одним из главных условий успешной реализации больших ПП. Однако согласно мнению ведущего специалиста компании Construx Software Builders Inc Макконнелла (S. С. McConnell): «Даже при наличии квалифицированных, мотивированных и трудолюбивых людей неверная структура команды способна свести на нет их усилия, вместо того чтобы привести их к успеху. Слабая структура команды может послужить причиной увеличения времени разработки, снижения качества, понижения морального духа, повышения текучести кадров и, в конечном итоге, привести к провалу проекта».

При формировании профессиональной команды IT-разработчиков ПП большую роль играет руководитель, причем известные крупные компании особо подчеркивают сложность проблемы поиска приемлемой кандидатуры руководителя проекта.

Важную роль при работе над проектом играет подготовка и обучение разработчиков, а также наличие хороших коммуникаций в их команде (отсутствие может привести к появлению у некоторых членов команды чувства отстраненности и безразличия к проекту).

Проблемы выбора программных средств разработки ПП. Современная тенденция автоматизации разработки больших ПП связана

с широким использованием CASE-средств таких, например, как Microsoft Project, Primavera Project Planner, BPwin и др. По мнению вице-президента компании PSM Consulting Ньюэлла (M. Newell), «... программное обеспечение для управления проектами при правильном использовании может стать одним из самых ценных ваших инструментов». Однако на практике возникает естественная проблема оценки и выбора этих средств, т. е. оценки их функциональности и качества: надежность, переносимость и др.

В общем случае выбор CASE-средств для реализации ПП зависит от целей, потребностей и ограничений проекта [14]. Определяющим фактором такого выбора является используемая методология проектирования, причем оценку и выбор следует начинать только после тщательного анализа количественных и качественных требований к проекту.

Проблемы анализа жизнеспособности ПП. Приступая к реализации больших ПП, необходимо, прежде всего, понять целесообразность проекта и оценить оправданность вложения в него трудовых и материальных затрат, т. е., как отмечает В. В. Липаев, «... провести оценку реализуемости проекта в условиях и ресурсах, предлагаемых заказчиком» [9, 15]. Согласно мнению директора консалтингового департамента КГ СЭТ И. Абрамовой: «... следствием перманентного цейтнота являются не только стрессы сотрудников, но и практически полное отсутствие подготовительного этапа работы над проектом».

Оценка реалистичности разработки ПП, именуемая часто как реализуемость (В. В. Липаев), осуществимость (С. Орлик, И. Соммервилл (I. Sommerville)) или жизнеспособность (И. И. Мазур, А. Е. Колоденкова) проекта, позволяет спрогнозировать успешность его выполнения еще на концептуальной стадии.

Однако такой прогноз существенно усложняется наличием следующих факторов:

- Неполнота исходных данных для проектирования, не позволяющая дать точную оценку жизнеспособности проекта и, прежде всего, срокам его выполнения. Здесь следует отметить часто имеющее место игнорирование накопленного опыта разработчиков: показатели удачно реализованных проектов не обобщаются и не используются в качестве исходной базы для управления новыми проектами. Как заметил ДеМарко (T. DeMarco) [16], «Риски, которые мы должны в первую очередь анализировать в новом проекте, это проблемы проектов вчерашних».

- Изменение требований, сроков и объема выделяемых ресурсов на проектирование, прямо связанное с увеличением риска осуществления проекта с должным качеством. Согласно Бюэму (В. Boehm) «непрекращающийся поток изменений» является одним из наиболее распространенных рисков программного проекта. Здесь, пожалуй, можно согласиться с И. Абрамовой, утверждающей, что «превышение выделенного бюджета и цейтнот в конце проекта часто связаны с тем, что в процессе работы ставятся все новые и новые задачи, не предусмотренные первоначальным планом».

- Слабая структурированность теоретических и фактических знаний о проекте, не позволяющая дать удовлетворительный прогноз рисков проекта на достаточно большом промежутке времени с применением неформализованных методов на основе лишь собственного опыта и интуиции. В связи с этим, известный специалист по управлению проектами А. Коберн (А. Cockburn) отмечает: «... в разработке ПО проблема заключается в том, что при постановке задачи и проектировании допускается слишком много неточностей. Все будет хорошо, если мы сможем заставить людей работать с математическим формализмом».

Разработка любого большого ПП сопровождается генерацией (разработкой, формированием) альтернатив его реализации, которые сравниваются друг с другом с целью выбора из них наилучшего. Чем больше будет разработано альтернатив, тем выше будет обоснованность принимаемых решений по жизнеспособности проекта. При этом рассмотрение слишком большого числа альтернатив ведет к большим усилиям, затратам времени и путанице. Поэтому руководители, как правило, ограничиваются рассмотрением всего нескольких, наиболее жизнеспособных альтернатив.

Для обеспечения большей реалистичности оценки жизнеспособности разрабатываемого ПО имеющиеся альтернативы следует оценивать многокритериально. При этом, естественно, встает весьма трудный вопрос о полноте списка критериев. Уточнение критериев отбора альтернатив, основанного на детально проработанных вариантах реализации проекта, возлагается на руководителя проекта.

Эффективное управление ПП и повышение качества разрабатываемого ПО на этапе анализа жизнеспособности проекта достигается за счет применения методов принятия решений (мате-

матических, эвристических), которые могут быть различными в зависимости от типа решаемых задач или проблем [13].

Заметим, что многие специалисты в области разработки и управления сложными ПП (Е. Поваляев, А. С. Волков, В. Е. Гвоздев, Б. Г. Ильясов) подчеркивают необходимость применения на начальных стадиях проекта моделей процесса разработки ПО (часто при описании процессов вместо слова модель употребляется термин методология), с целью лучшего понимания разрабатываемого ПО (см., например [14, 28]). Модели позволяют свести высокую сложность проектов до уровня, понимаемого человеком. Достигается это за счет иерархического принципа их построения и применения наглядной графической нотации. Каждая модель может быть выражена с разными уровнями детализации.

#### **4. КРИТЕРИИ И ПОКАЗАТЕЛИ ЖИЗНЕСПОСОБНОСТИ ПРОГРАММНЫХ ПРОЕКТОВ**

Для оценки жизнеспособности ПП предлагается классификация видов, критериев и показателей жизнеспособности проекта, представленная на рис. 3 [6].

Виды жизнеспособности проекта. В процессе планирования и организации инновационной деятельности проводят оценку жизнеспособности проекта, которая включает следующие виды его анализа: технический (рассмотрение альтернативных вариантов реализации проекта и оценки их реализуемости, сроков осуществления проекта в целом и всех его стадий, составление календарных планов), финансовый (определение соотношения финансовых затрат и результатов, обеспечивающих необходимую норму доходности), экономический (отражение эффективности проекта с точки зрения интересов всего общества в целом, например, поступлений средств в различные бюджеты), коммерческий (определение каналов продвижения программных продуктов на рынок; анализ и оценка конкурентов).

Критерии жизнеспособности проекта. Выбор проектов с помощью критериев направлен на выявление общего представления о проекте (его достоинствах и недостатках). При составлении перечня критериев необходимо использовать лишь те из них, которые соответствуют приоритетным целям и задачам.



Рис. 3. Классификация видов, критериев и показателей жизнеспособности ПП

К критериям выбора проектов относятся: научно-технические (перспективность используемых научно-технических решений, применения полученных результатов), производственные (данные о наличии персонала, доступности сырья, материалов), финансово-экономические (ожидаемая норма прибыли, срок окупаемости), маркетинговые (оценка рыночного потенциала).

Показатели жизнеспособности проекта. Критерии выбора проектов могут оцениваться прямыми и косвенными показателями. Прямые показатели непосредственно характеризуют критерий жизнеспособности проекта (время разработки, количество исполнителей и потребителей и др.). Косвенные показатели используются в том случае, когда невозможно получить значения прямых показателей проекта (емкость рынка, стоимость работ, вероятность успеха проекта и т. д.).

Следует особо подчеркнуть многокритериальность понятия жизнеспособности ПП. Действительно, квалифицированная оценка жизнеспособности проекта  $\pi$  является векторной и предполагает достаточную широту охвата учитываемых показателей жизнеспособности (как прямых, так и косвенных), которые обозначим унифицировано через  $\pi_1, \pi_2, \dots, \pi_n$ :

$$\pi = (\pi_1, \pi_2, \dots, \pi_n),$$

а их заданные значения – через  $\pi_1^*, \pi_2^*, \dots, \pi_n^*$ :

$$\pi^* = (\pi_1^*, \pi_2^*, \dots, \pi_n^*).$$

Тогда требования, предъявляемые к жизнеспособности проекта, определяются его функциональным назначением. При этом возможны следующие три способа формулировки данных требований [11]:

- в виде допустимого уровня жизнеспособности проекта, т. е. в требовании не превышения показателей жизнеспособности их заданных допустимых значений:

$$\pi_i \leq \pi_i^*, \quad i = \overline{1, n};$$

- в виде заданного уровня жизнеспособности проекта, т. е. в требовании точного или приближенного равенства показателей жизнеспособности их заданных допустимых значений:

$$\pi_i = \pi_i^*, \quad i = \overline{1, n};$$

- в виде оптимального уровня жизнеспособности проекта, т. е. в требовании экстремальности (возможной предельности) значений показателей жизнеспособности проекта:

$$\pi_i \rightarrow \text{extr}, \quad i = \overline{1, n}.$$

Следует заметить, что на протяжении всей истории развития программного инжиниринга отмечалось стремление к предельному совершенствованию тех или иных показателей жизнеспособности ПП, следуя принципу: «Мы не настолько богаты, чтобы проектировать неоптимально». В связи с этим наибольшую популярность в практике ПП получают именно постановки задачи обеспечения оптимальных

(максимальных, либо минимальных в зависимости от особенностей проекта) показателей жизнеспособности проекта.

### 5. ЗАДАЧИ И МЕТОДЫ ОЦЕНКИ ЖИЗНЕСПОСОБНОСТИ ПРОГРАММНЫХ ПРОЕКТОВ

На разных стадиях и этапах жизненного цикла ПП возникает широкий спектр задач оценки жизнеспособности проекта, которые являются следствием манифеста программной инженерии: «максимизация качества проекта при минимуме затрат». Ясно, что успешное решение данных задач обеспечивает успешное выполнение проекта.

В общем случае задача оценки жизнеспособности ПП является задачей весьма сложной и современная программная инженерия предлагает все новые подходы к ее решению, однако до сих пор нет надежной универсальной методики, дающей гарантированный результат. Как подчеркивает Ф. Брукс: «... слабо развиты наши методы оценок. В сущности, они отражают молчаливое и совершенно неверное предположение, что все будет идти хорошо». На рис. 4, на основе систематизации обзора литературных источников, представлена классификация существующих задач оценки жизнеспособности ПП.

Задачи исследования операций, связанные с поиском путей рационального использования имеющихся ресурсов для реализации поставленной цели проекта. Здесь можно выделить следующие задачи:

- задачи управления запасами, связанные с созданием запаса материальных ресурсов или предметов потребления при проектировании с целью удовлетворения спроса на заданном интервале времени;
- задачи распределения ресурсов, связанные с необходимостью выполнения определенного набора работ при проектировании с учетом ограниченных ресурсов;
- задачи упорядочивания, связанные с формированием очередности работ при проектировании в порядке их значимости.

Задачи программного инжиниринга, связанные с оценкой жизнеспособности проектов в условиях ограниченных временных ресурсов. Здесь для большинства проектов ключевыми являются следующие оптимизационные задачи:

Минимизация упущенной выгоды [17, 18]. В связи с тем, что в условиях дефицита финансовых ресурсов нет возможности вести одновременно календарные планы реализации всех проектов, необходимо выбрать первоочередные

проекты, реализация которых обеспечивает наибольший эффект. Реализация остальных проектов отодвигается на более поздние сроки, и ставится задача разработать план реализации программы, при котором упущенная выгода минимальна.

Например, в [17] рассматривается задача распределения ресурсов по проектам таким образом, чтобы минимизировать упущенную выгоду. Для ее решения предложен эвристический алгоритм, в основе которого лежит упорядочение проектов по убыванию приоритетов  $q_i = c_i / w_i$  ( $w_i$  – объем финансирования,  $c_i$  – доход после завершения проекта).

В работе [18] рассматривается задача определения продолжительности всех операций таким образом, чтобы минимизировать упущенную выгоду  $\sum_{i=1}^n c_i t_i$  ( $n$  – количество независимых операций) при ограничениях на суммарные затраты  $\sum_{i=1}^n s_i(t_i) \leq S$  ( $s_i(t_i)$  – зависимость затрат для каждой операции от продолжительности ее выполнения  $t_i$ ). Для решения данной задачи применяют метод множителей Лагранжа. Следует заметить, что если ограничение выполняется, то задача элементарна, иначе проект не реализуем.

Минимизация проектов по стоимости и срокам [19, 20]. Общая стоимость проекта зависит от стоимости выполнения каждой работы, а также от любых дополнительных переменных или постоянных расходов. Снижение продолжительности выполнения некоторых работ можно осуществлять с помощью дополнительных ресурсов. В связи с этим ставится задача минимизации стоимости и сроков выполнения проекта.

К примеру, в [20] рассматривается задача нахождения такого значения  $K$ , чтобы общая длительность проекта сократилась по сравнению с «нормальным» (сначала все требования будут собраны, а затем реализованы) выполнением. Причем решение задачи предполагает предварительную оценку сложности проекта. Данная задача сводится к решению следующей системы двух неравенств:

$$(N-K)\tau_Y > (\tau_W \lambda (N-K)^2) / N;$$

$$K\tau_W > (N-K)\tau_Y,$$

где  $N$  – требования;  $(N - K)$  – оставшиеся требования;  $\tau_Y$  – время аналитика, требующееся на выявление одного требования;  $\tau_W$  – время программиста, требующееся на выявление одного требования;  $\lambda$  – коэффициент «усложнения».



Рис. 4. Классификация задач оценки жизнеспособности ПП

Следует отметить, что если у второй системы неравенств существование решения ограничено многими условиями, то у первой оно существует при большом количестве возможных значений параметров.

Минимизация затрат при формировании команды исполнителей [21, 22]. Поскольку скорость вывода новых программных продуктов на рынок в современных условиях является конкурентным преимуществом, то естественно возникает задача формирования такой команды исполнителей проекта, которая бы осуществила выполнение всех работ в заданные сроки, и при этом затраты на оплату труда были минимальны.

Например, в работе [21] рассматривается задача минимизации затрат при формировании команды, описываемая следующей целевой функцией и ограничениями:

$$\sum_{i=1}^n a_i x_i \rightarrow \min, \quad \sum_{i=1}^n b_i x_i \geq P$$

( $a_i$  – заработная плата  $i$ -го специалиста,  $b_i$  – индивидуальная производительность;  $x_i$  – двоичная переменная (если  $i$ -й специалист принимает участие в реализации проекта  $x_i = 1$  и  $x_i = 0$  в противном случае);  $P$  – минимально необходимая производительность труда для выполнения проекта в заданные сроки ( $P = Q / T$ , где  $Q$  –

трудоемкость выполняемого проекта,  $T$  – сроки выполнения)). Предложенная задача представляла собой классическую задачу о «ранце» (класс задач линейного программирования), для решения которой используется метод ветвей и границ.

Ранжирование работ, проектов [23, 24]. В условиях ограниченности финансовых ресурсов для предприятий крайне важно реализовывать наиболее эффективные проекты, поэтому на первом этапе необходимо выстроить проекты в порядке убывания их значимости для того, чтобы далее производить отбор. В связи с этим необходимо разрабатывать методики, в которых были бы прописаны показатели и принципы, на основании которых осуществляется ранжирование.

В [23] рассматривается задача выбора наиболее приоритетных работ в рамках проекта

$$U(M) = \sum_{i \in M} u_i,$$

где  $M$  – подмножество работ,  $u_i$  – полезность  $i$ -й работы при заданном ограничении на время выполнения общего набора работ

$$\sum_{i \in M} t_i \leq T,$$

где  $t_i$  – время выполнения  $i$ -й работы,  $T$  – общее время выполнения проекта. Для решения дан-



ной задачи используются эвристические алгоритмы метода дискретной оптимизации с ранжированием полезности работ.

В работе [24] рассматривается задача отбора приоритетных проектов как задача многокритериальной дискретной оптимизации

$$(f_1(j), f_2(j), \dots, f_p(j)) \rightarrow \max, j \in X,$$

где  $X = \{1, 2, \dots, j, \dots, m\}$  – множество допустимых альтернатив,  $f_1, f_2, \dots, f_p$  – критерии качества проекта;  $y^j = (f_1(j), f_2(j))$  – векторная оценка каждого  $j$ -й проекта. Для решения данной задачи было разработано два подхода: первый подход основан на поиске всех Парето-оптимальных решений; второй подход – на задаче упорядочения всех проектов в порядке убывания «предпочтительности».

Безусловно, что указанные задачи могут быть положены в основу технологии компьютерной оценки параметров жизнеспособности ПП и принятия решений по дальнейшему управлению проектом.

#### **6. ВЕРОЯТНОСТНО- СТАТИСТИЧЕСКИЕ ПОДХОДЫ К ОЦЕНКЕ ЖИЗНЕСПОСОБНОСТИ ПРОГРАММНЫХ ПРОЕКТОВ**

Исторически первыми появились вероятностно-статистические подходы и на сегодняшний день они являются наиболее развитыми. Эти методы описания и анализа неопределенности являются основой для принятия решений по жизнеспособности проекта в условиях риска. Суть вероятностно-статистических методов заключается в использовании вероятностных моделей на основе оценивания и проверки гипотез с помощью выборочных характеристик [25].

Следует заметить, что непосредственные усилия разработчиков (трудоzатраты) определяют значительную часть стоимости разработки программного продукта и методы их анализа дают оценки, которые впоследствии могут быть преобразованы в продолжительность или стоимость проекта. В настоящее время при анализе трудоzатрат проекта широко применяются экспертные оценки. Однако данный подход сопряжен с рядом проблем: основания для получения оценки не являются явными; трудно находить квалифицированных экспертов для оценки каждого нового проекта и др. В связи с этим было разработано множество различных моделей анализа трудоzатрат, которые варьируются от моделей, основанных на эмпирических данных (модель СОСОМО [6, 9, 26]), до чисто аналитических моделей. Эмпирические модели исполь-

зуют данные предыдущих проектов для оценки текущих (анализируя закономерности с помощью имеющихся баз данных предыдущих проектов).

Метод «СОСОМО II». При построении СОСОМО II для обработки статистических данных используется байесовский анализ, который дает лучшие результаты для программных проектов, характеризующихся неполнотой и неоднозначностью. Также в модели допускается измерять размер проекта не только числом строк кода (KSLOC, Kilo Source Lines Of Code), но и более современными функциональными и объектными точками. Размер программного продукта может быть, например, оценен экспертами с применением метода PERT. Формула оценки трудоzемкости проекта имеет следующий вид [6]:

$$PM = 2,74 \cdot SIZE^E \cdot \prod_{i=1}^n EM_i ;$$

$$E = 0,91 + 0,01 \cdot \sum_{j=1}^5 SF_j ,$$

где  $SIZE$  – размер продукта в KSLOC;  $EM_i$  – множители трудоzемкости;  $SF_j$  – факторы масштаба;  $n = 7$  – для предварительной оценки;  $n = 17$  – для детальной оценки.

Однако, несмотря на отмеченные достоинства, модель имеет ряд недостатков:

- методика СОСОМО II рассчитана только на относительно маленькие проекты, разрабатываемые командами, хорошо знакомыми с прикладной областью;
- необходимо знать размер программного продукта в тысячах строках исходного кода, который в свою очередь зависит от языка программирования [27].
- сложные формулы, как правило, очень чувствительны к точности большого числа параметров (в приведенном примере формул СОСОМО II содержится 21 параметр), которые необходимо задать, чтобы получить требуемые оценки.

В литературе по программному инжинирингу подчеркивается, что одним из условий повышения эффективности оценки жизнеспособности ПП является использование формально математических методов, без которых дать удовлетворительный прогноз развития проекта на достаточно большом промежутке времени, опираясь только лишь на собственный опыт и интуицию, практически невозможно. В настоящей работе для оценки жизнеспособности программных проектов предложено использо-

вать вероятностно-статистический подход, основанный на широко распространенной в современной науке функциональной парадигме [28], согласно которой все необходимые компоненты разрабатываемых программных средств представляют собой «черные ящики», выполняющие сформулированные в требованиях задачи функции при отсутствии описания их внутренней реализации.

Метод «Обращение-Сдвиг». В основу предлагаемого подхода положено использование системной модели оценки жизнеспособности ПП, представленной на рис. 5 [2–4].

Как видно, жизнеспособность реализации проекта определяется, с одной стороны, требованиями, которым ставится в соответствие объем работ по проекту; с другой стороны, существующими ограничениями на ресурсы (время выполнения проекта); с третьей стороны, производительностью исполнителей.

При постановке задач ресурсного планирования предполагается, что проект описывается в виде сетевого графика выполнения работ. При этом в силу неоднозначности выделения работ одному и тому же проекту можно поставить в соответствие различные варианты сетевых графиков. Предполагается, что любой такой сетевой график можно представить в виде последовательно-параллельных схем выполнения работ, которые в графиках равнозначны.

Рассмотрим вероятностно-статистический подход к оценке жизнеспособности ПП, предложенный автором в работе [2–4]. Идея подхода заключается в сравнительном анализе множества альтернатив сетевых графиков выполнения работ в рамках проекта на основе специально сформированного скалярного статистического показателя его жизнеспособности, который для  $i$ -го варианта проекта представляется в виде ве-

роятности выполнения совокупности работ за выделенное время:

$$\alpha^{(i)} = \min\{p_j^{(i)}\}, \quad i = \overline{1;m}, \quad j = \overline{1;k_i},$$

где  $m$  – число вариантов реализации проекта,  $k_i$  – количество видов работ в  $i$ -м варианте реализации проекта,  $p_j^{(i)}$  – вероятность выполнения  $j$ -й работы в  $i$ -м варианте реализации проекта, представленных на  $i$ -м сетевом графике при заданных ограничениях на время  $T_j^{(i)}$  выполнения каждой работы. При этом полагаем, что вероятность  $p_j^{(i)}$  зависит от времени  $T_j^{(i)}$ , а также от исполнителей  $j$ -й работы  $i$ -го варианта реализации проекта. Данный подход позволяет на стадии анализа жизнеспособности проекта принять решение об отказе от анализируемой альтернативы проекта, о запуске его в производство либо о генерации новых альтернатив реализации проекта.

Пусть задана производительность исполнителей  $T_n = \psi_{j,n}(V_j^{(i)})$  ( $T_n$  – время выполнения работ  $n$ -м разработчиком),  $i = \overline{1;m}$ ,  $j = \overline{1;k_i}$ , а также варианты сетевых графиков выполнения работ с указанием объема для каждой работы  $V_j^{(i)}$ . Для расчета скалярной характеристики  $\alpha^{(i)}$ , характеризующей жизнеспособность  $i$ -го альтернативного варианта реализации проекта, предлагается алгоритм, который содержит следующие четыре этапа:

- на первом этапе рассчитываются ожидаемые времена выполнения работ  $T_j^{(i)}$  для различных  $m$  вариантов сетевых графиков выполнения работ с указанными объемами работ  $V_j^{(i)}$  и производительностями исполнителей;

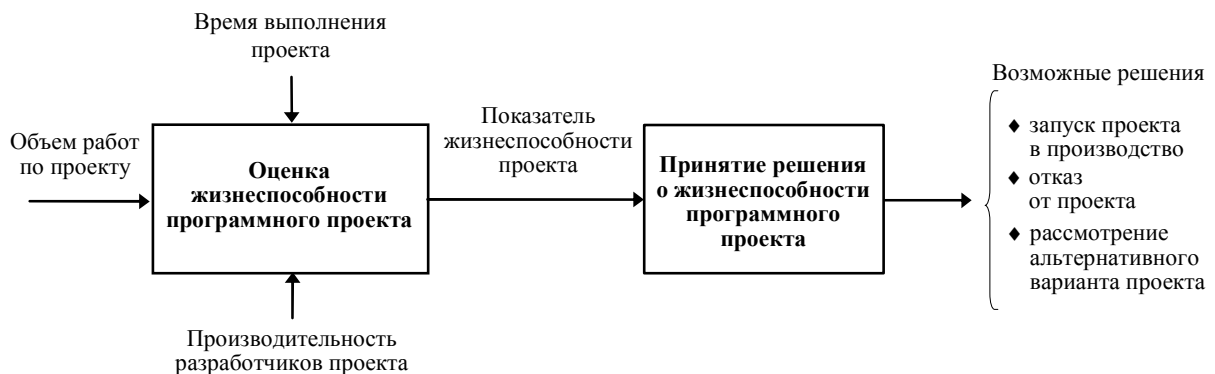


Рис. 5. Системная модель оценки жизнеспособности ПП

- на втором этапе рассчитываются вероятности выполнения работ  $p_j^{(i)}$  для различных  $m$  вариантов сетевых графиков с ожидаемыми временами выполнения работ  $T_j^{(i)}$ ;

- на третьем этапе формируется скалярная характеристика жизнеспособности программного проекта  $\alpha^{(i)}$  для каждого  $i$ -го альтернативного варианта реализации проекта, которая определяется минимальной вероятностью выполнения работ  $\alpha^{(i)} = \min_i \{p_j^{(i)}\}$ ;

- на четвертом этапе осуществляется принятие решений на основе анализа множества характеристик  $\alpha^{(i)}$ ,  $i = \overline{1; m}$ .

Обратимся к первым двум этапам алгоритма решения задачи анализа жизнеспособности проекта. Ожидаемые времена выполнения работ и вероятности выполнения для каждой работы рассчитываются по распределениям вероятности объема, времени окончания работ и производительности исполнителей проекта. Последние находятся на основе известных архивных данных, полученных по предыдущим аналогичным проектам, которые, как правило, собираются, накапливаются и обрабатываются в проектных организациях. При этом следует отметить, что для нахождения производительности исполнителей проекта используется непараметрический метод, описанный в работе автора [29].

В контексте рассматриваемой задачи, роль времени выполнения работ  $T_j^{(i)}$  играет случайная величина  $X$ , а объемов работ  $V_j^{(i)}$  – случайная величина  $Y$ , значения которых будем обозначать соответственно через  $x$  и  $y$ . Считаем, что производительность исполнителей  $T_n = \psi_{j,n}(V_j^{(i)})$ , т. е. зависимость  $y = \varphi(x)$  – известная монотонная, возрастающая функция.

Тогда нетрудно видеть, что первые два этапа алгоритма решения поставленной задачи сводятся к нахождению по имеющимся выборочным значениям  $y$  случайной величины  $Y$  и известной функциональной зависимости  $y = \varphi(x)$  закона распределения  $F_1(x)$  случайной величины  $X$ . Данная задача, являющаяся обратной задачей по отношению к известной классической прямой задаче оценивания закона распределения функции случайного аргумента,

в литературных источниках не рассматривалась. Для ее решения, т. е. построения искомого закона распределения  $F_1(x)$  случайной величины  $X$ , предлагается использовать метод «Обращение-Сдвиг», предложенный автором в работе [2–4] и состоящий из следующих шести шагов (графическая иллюстрация метода представлена на рис. 6).

На первом шаге рассчитывается математическое ожидание  $M[y]$  и среднее квадратическое отклонение  $\sigma[y]$  величины  $Y$  по ее выборочным значениям  $y$ .

На втором шаге рассматривается условие  $M[y] / \sigma[y] \geq 3,5$ , при выполнении которого осуществляется сдвиг функции  $z_2 = F_2(y)$  (кривая 1) на величину  $d$ .

На третьем шаге осуществляется построение функции  $z_2^* = F_2(y^*)$  (кривая 2) с учетом сдвига по преобразованным данным  $y_i^* = y_i - d$  ( $i = \overline{1; N}$ ,  $N$  – объем выборки) и осуществляется сдвиг функции  $y = \varphi(x)$  (кривая 3) на величину  $d$ .

На четвертом шаге формируется выборка  $\{y_p^*\}$  на основе функции  $z_2^* = F_2(y^*)$  (кривая 2). При достаточно малом шаге  $\lambda_p \in [0; 1]$ ,  $p = 1, 2, \dots$  изменения значения  $\lambda$  функции распределения  $F_2(y^*) \in [0; 1]$  можно получить выборку  $\{y_p^*\}$  достаточно большого объема.

На пятом шаге формируется выборка  $\{x_p\}$  посредством обращения функции  $y^* = \varphi(x) - d$  (кривая 4).

На шестом шаге осуществляется построение искомого функции  $z_1 = F_1(x)$  (кривая 5) по выборке  $\{x_p\}$ .

Относительно параметра сдвига  $d$ , используемого на втором и третьем шагах метода, следует заметить, что при построении функции  $z_2 = F_2(y)$  (кривая 1) может возникнуть «хвост» интегральной функции распределения [30], которым для упрощения вычислений целесообразно пренебречь, т.е. преобразовать исходные данные  $\{y_i\}$  на величину сдвига  $d = M[y] - 3,5 \sigma[y]$ . При этом значения  $y$  случайной величины  $Y$  предлагается рассматривать внутри интервала  $3,5\sigma$ , поскольку вероятность нахождения их вне его весьма мала.

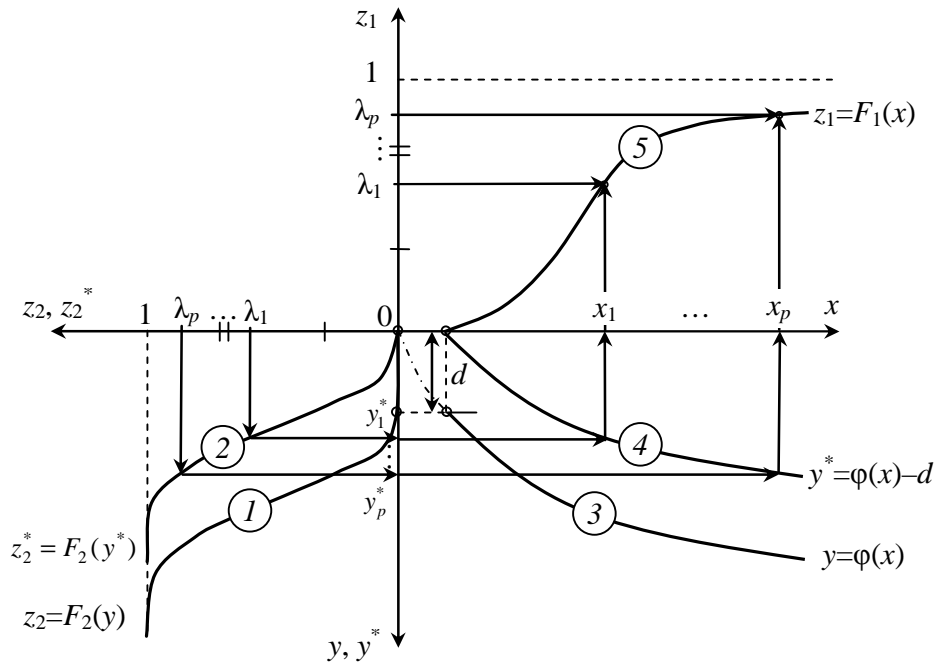


Рис. 6. Графическая иллюстрация метода «Обращение-Сдвиг»

Здесь мы рассмотрели частный случай задачи нахождения функции распределения  $F_1(x)$  с учетом сдвига. Однако описанный подход к формированию выборок  $\{y_p\}$ ,  $\{x_p\}$  применим и для общего случая, когда условие  $M[y] / \sigma[y] \geq 3,5$  не выполняется и сдвиг функции  $z_2 = F_2(y)$  (кривая 1) нецелесообразен.

### 7. ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ АНАЛИЗА ЖИЗНЕСПОСОБНОСТИ ПРОГРАММНЫХ ПРОЕКТОВ

Современная программная инженерия приблизилась к границе, за которой, по выражению Негойцэ (С. V. Negoita), «... существенную роль начинают играть способы учета неопределенностей», т.е. так называемых НЕ-факторов внешней и внутренней среды ПП, отражающих неполноту знаний, их недостоверность, а также нечеткость и неточность, относящихся к их содержанию. Очевидно, что для преодоления данной неопределенности необходим неформальный акт, связанный с привлечением тех или иных гипотез поведения факторов, порождающих неопределенность.

В настоящее время существуют различные конкурирующие гипотезы, опираясь на которые, удастся «устранить» неопределенность и придать проблеме управления ПП количественную определенность. Среди различных способов формализации неопределенности наибольшее распространение получил стохастический (ве-

роятностно-статистический) подход и, в частности, описанный выше подход, основанный на методе «Обращение-Сдвиг», позволяющий в терминах случайности моделировать неполноту знаний об объеме работ и производительности исполнителей проекта. Данный подход является весьма эффективным при анализе жизнеспособности, например, ПО для информационной поддержки контроля и управления состоянием территориальных систем [30]. Дело в том, что в силу многомерности и многосвязности данных систем, а также недостаточной изученности протекающих в них процессов, при их изучении особое место занимают именно статистические модели, ориентированные на обработку малых по объему и низким по точности исходных данных.

Однако на пути широкого распространения стохастических моделей неопределенности часто возникают затруднения, вызванные некорректным либо неправомерным использованием методологии теории вероятностей и математической статистики [31]. Как утверждает Касты (J. Casti): «... нет априорных математических оснований полагать, что механизм, порождающий неопределенность, по своей природе непременно стохастичен». В литературе общепризнанно считать неопределенными величины, для которых не обнаруживается статистическая устойчивость. Это обстоятельство явилось основанием для высказываний Калмана

(R. E. Kalman): «Мы должны отрицать, что классические вероятностные структуры классической теории вероятностей, на самом деле, имеют научное отношение к описанию неопределенности», а также Н. Н. Моисеева: «Стохастические задачи, т.е. задачи, содержащие случайные величины или функции, мы не относим к числу задач, содержащих неопределенные факторы». В связи с этим, в последние годы все большее распространение находят различные интервальные модели [32], а также нестохастические модели неопределенности типа (см., например, [33, 34]: субъективная вероятность Севеджа (L. J. Savage), верхняя и нижняя вероятности Демпстера (A. P. Dempster), правдоподобие и доверие Шеффера (G. Shafer), емкость Шоке (G. Choquet), возможности Заде (L. A. Zadeh), Шейкла (G. L. S. Shackle) и Ю. П. Пытьева и др.).

Весь арсенал используемых статистических методов можно распределить по трем потокам [35]: высокие, классические и низкие статистические технологии.

Несмотря на большое разнообразие подходов к оценке жизнеспособности ПП с использованием указанных моделей неопределенности, наиболее перспективными являются комплексные подходы, получившее название «высоких статистических технологий» [35], под которыми понимаются пять точек роста: непараметрическая статистика, робастность, бутстреп, статистика интервальных и нечисловых данных. Для задач оценки жизнеспособности ПП наиболее перспективным является статистика интервальных и нечисловых данных.

Следует отметить, что описанные выше подходы не отрицают классическую теорию вероятностей, а дополняют и расширяют ее.

### ЗАКЛЮЧЕНИЕ

Таким образом, жизнеспособность является важным этапом ЖЦ ПП. Попытки применения какого-либо конкретного подхода для принятия решений в условиях неопределенности позволяют отразить в модели лишь некоторые виды данных, а это в свою очередь приводит к потере информации других видов данных. Особенно это относится к большим ПП. В связи с этим дальнейшие исследования автора будут направлены на комбинирование различных формальных и неформальных подходов к концептуальному проектированию ПО.

### СПИСОК ЛИТЕРАТУРЫ

1. Официальный сайт Президента России, Дмитрий Медведев. 2010 URL: <http://archive.kremlin.ru>.
2. **Колоденкова А. Е.** Задачи программного инжиниринга сложных систем на основе критерия жизнеспособности проекта // Проблемы управления и моделирования в сложных системах: Тр. XII Междунар. конф. Самара: Самарский НЦ РАН, 2010. С. 593–598.
3. **Колоденкова А. Е.** Статистический подход к оценке реалистичности программных проектов для автоматизированных информационно-управляющих систем // Мехатроника, автоматизация, управление. 2010. № 4. С. 52–60.
4. **Колоденкова А. Е.** Анализ жизнеспособности – важная стадия жизненного цикла инновационных программных проектов // Программная инженерия. 2010. № 1. С. 18–25.
5. **Макконнелл С.** Сколько стоит программный проект. М.: Русская редакция; СПб.: Питер, 2007. 304 с.
6. **Архипенков С. Я.** Лекции по управлению программными проектами. М., 2009. URL: [http://www.arkhipenkov.ru/resources/sw\\_project\\_management.pdf](http://www.arkhipenkov.ru/resources/sw_project_management.pdf).
7. **Мандрикова Л. В., Манжос Ю. С., Лучшев П. А.** Методы оценки стоимости и затрат на создание программного продукта, основанные на нечеткой логике // Радиоэлектронные и компьютерные системы. 2008. №2. С. 115–118.
8. **Мандрикова Л. В., Манжос Ю. С., Хоменко В. В.** Метод идентификации рисков программного проекта на основе вероятностного подхода // Радиоэлектронные и компьютерные системы. 2009. № 7. С. 207–211.
9. **Липаев В. В.** Программная инженерия. Методологические основы: учеб. М.: ГУВШЭ, ТЕИС. 2006. 608 с.
10. Standish Group Report: CHAOS Summary 2009. URL: [http://www.standishgroup.com/newsroom/chaos\\_2009.php](http://www.standishgroup.com/newsroom/chaos_2009.php).
11. **Чуйкин А. М.** Разработка управленческих решений: учеб. пособие. Калининград: Калинингр. ун-т, 2000. 150 с.
12. **Мескон М., Альберт М., Хедоури Ф.** Основы менеджмента. М.: Дело, 1997. 502 с.
13. **Колоденкова А. Е.** Проблемы концептуального проектирования программного обеспечения мехатронных систем // Мехатроника, автоматизация, управление: Матер. 7-й науч.-техн. конф. С.-Петербург: ГНЦ РФ ЦНИИ «Электроприбор», 2010. С. 228–231.
14. **Гвоздев В. Е., Колоденкова А. Е.** Системные вопросы проектирования программных продуктов: учеб. пособие. Уфа: Уфимск. гос. авиац. техн. ун-т, АН РБ, Гилем, 2010. 188 с.

15. **Липаев В. В.** Проблемы программной инженерии: качество, безопасность, риски, экономика // Программная инженерия. 2010. № 1. С. 7–20.
16. **ДеМарко Т., Листер Т.** Вальсируя с Медведями: Управление рисками в проектах по разработке программного обеспечения. М.: Компания р.п. Office, 2005. 208 с.
17. **Бурков В. Н., Квон О. Ф., Цитович Л. А.** Модели и методы мультипроектного управления. М.: ИПУ РАН, 1997. 62 с.
18. **Баркалов С. А., Портных В. А., Семенов П. И.** Минимизация упущенной выгоды в случае  $n$  независимых операций // Теория активных систем: Тр. Междунар. науч.-практич. конф. Москва, 2001. Т. 1. С. 21–22.
19. **Романов В. С.** Задача управления стоимостью компании – дискретный случай // Управление большими системами. М.: ИПУ РАН, 2006. С. 142–152.
20. **Маркаров Д. А.** Частная задача оптимизации сроков при управлении проектами // Управление большими системами. М.: ИПУ РАН, 2005. С. 53–59.
21. **Галинская Е. А., Половинкина А. И., Рентеева Е. Л.** Модель минимизации затрат при формировании команды информационного проекта // Управление большими системами. Вып. 14. Воронеж: ВГАСУ, 2006. С. 29–33.
22. **Новиков Д. А.** Модели формирования и функционирования неоднородных команд. М.: ИПУ РАН, 2003. С. 73–90.
23. **Бекмурзаев В. А., Шеховцов А. А.** Использование метода динамического программирования и его оптимизация при решении задач управления проектами // Журнал научных публикаций аспирантов и докторантов. 2008. URL: <http://www.jurnal.org/articles/2008/inf18.html>.
24. **Черникова А. А.** Формирование инвестиционных пакетов, удовлетворяющих целям регионального развития // Инновационный Вестник Регион. 2007. № 2. С. 38–41.
25. **Орлов А. И.** Теория принятия решений. М.: Экзамен, 2005. 656 с.
26. **Boehm В.** Cocomo II model definition manual. Computer Science Department, University of Southern California, 1997. 540 p.
27. **Орлов С. А.** Технологии разработки программного обеспечения. СПб.: Питер, 2002. 464 с.
28. **Гвоздев В. Е., Ильясов Б. Г., Колоденкова А. Е.** Мехатроника, автоматизация, управление: Матер. 7-й науч.-техн. конф. С.-Петербург: ГНЦ РФ ЦНИИ «Электроприбор», 2010. С. 128–131.
29. **Гвоздев В. Е., Колоденкова А. Е.** Непараметрическое оценивание функциональных зависимостей по эмпирическим данным // Мехатроника, автоматизация, управление. 2005. № 8. С. 12–18.
30. Статистическое исследование территориальных систем / М. Б. Гузаиров [и др.]. М.: Машиностроение, 2008. 187 с.
31. **Филимонов Н. Б.** Смена парадигм неопределенности в задачах управления // Интеллектуальные системы: Тр. 6-го междунар. симпозиума М.: РУСАКИ, 2004. С. 173–178.
32. **Яценко Б. Н.** Оценка эффективности инвестиционных проектов и принятие инвестиционных решений в условиях большой неопределенности интервального типа // Аудит и финансовый анализ, 2006. № 1. С. 167–180.
33. **Клементьева С. В.** Применение теории нечетких множеств для измерения и оценки эффективности реализации наукоемкой продуктовой инновации // Заводская лаборатория. Диагностика материалов. 2006. № 11. С. 65–69.
34. **Пытьев Ю. П.** Возможность. Элементы теории и применения. М.: УРСС, 2000. 192 с.
35. **Орлов А. И.** Высокие статистические технологии // Заводская лаборатория. 2003. Т. 69. № 11. С. 55–60.

#### ОБ АВТОРЕ

**Колоденкова Анна Евгеньевна**, доц., зам. зав. каф. автоматиз. проектир. информ. систем. Дипл. инж.-системотехн. по АСОИУ (УГАТУ, 2004). Канд. техн. наук (УГАТУ, 2007). Иссл. в обл. программной инженерии, системн. анализа и прикл. статистики.