

Бахиркин Михаил Васильевич

На правах рукописи

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ
ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ЦИКЛА РАЗРАБОТКИ
ПРОГРАММНЫХ СИСТЕМ

Специальность 05.13.01 - Системный анализ, управление и обработка
информации (авиационная и ракетно-космическая техника)

Диссертации на соискание учёной степени
кандидата технических наук

Научный руководитель:
к. ф.-м. н., доцент, Лукин В. Н.

Москва, 2015

ОГЛАВЛЕНИЕ

Общая постановка задачи	8
Глава 1. Анализ успешности проектов	9
1.1 Определение проекта.....	9
1.2 Управление проектом.....	10
1.3 Стандарты в области управления проектами.....	12
1.4 Российские национальные стандарты в области управления проектами	13
1.5 Риски проектов и первичная оценка	14
1.6 Основные причины провала ИТ - проектов:	15
1.7 Анализ успешности проектов	15
1.8 Анализ успешности проектов в России	20
1.9 Анализ основных методов и метрик разработки программных проектов	21
Выводы по первой главе	49
Глава 2. Создания модели оценки.....	51
2.1. Формализация модели.....	51
2.2. Модель оценивания	51
2.3. Анализ оценок экспертов	51
2.4. Анализ распределения оценок экспертов.....	60
2.5. Построение математической модели	64
2.6. Алгоритм оценивания.....	67
2.7. Пример шага алгоритма оценивания	74
2.8. Программная реализация предложенного алгоритма	76
Выводы по второй главе	77
Глава 3. Использование предложенного метода в реальных проектах.....	79
3.1. Проект 1.....	79
3.2. Проект 2.....	83
3.3. Проект 3.....	88
3.4. Проект 4.....	89
3.5. Проект 5.....	89
3.6. Проект 6.....	89
3.6. Пример работы алгоритма ДОПС для третьего проекта	90
Выводы по третьей главе	91
Глава 4. Функциональные показатели процессов разработки ПО	92
4.1 Последовательные модели (Каскадная, водопада, V-образная)	94

4.2 Спиральная модель.....	95
4.3 Итеративные модели	97
4.4. Гибкие методологии.....	99
4.5 Уровень зрелости организации	102
Выводы по четвёртой главе	104
Заключение.....	106
Используемая литература	108

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ

▲ – начало доказательства.

▼ – окончание доказательства.

(F.1) – номер формулы.

Введение

Факт развития и продвижения в различные сферы человеческой деятельности информационных технологий представляется настолько очевидным, что наличие этого факта сейчас не обсуждают вовсе. Однако, качество предоставляемых услуг далеко не всегда на высоте, более того, возникает впечатление, что оно постепенно ухудшается. Часто “основной причиной этого считают увольнение квалифицированных специалистов-управленцев, увеличение срока и объёма работ и, как следствие, существенное усложнение процессов управления” [15]

В проектно-ориентированных организациях ежегодно выполняется множество ИТ-проектов, однако успешно завершаются только часть из них. Решение о перспективе проекта базируется на качественных и количественных оценках затрат, времени выполнения и прогнозируемой выгоде, т.е. качество проекта имеет решающее значение на стадии его инициализации.

Успех создания программной системы (ПС) определяется балансом таких факторов, как методология, персонал, границы проекта, время разработки, критерии качества. Если в ходе выполнения проекта что-то меняется в худшую сторону, то необходимо варьировать указанные факторы, чтобы выправить положение. Численность разработчиков увеличивать нельзя, как известно это только ухудшает ситуацию. Переход на новую методологию предполагает ее освоение. Сокращение границ проекта может помочь лишь тогда, когда из него исключаются еще не начатые работы, лежащие на критическом пути в сетевом графике. Но к концу проекта все работы должны находиться в стадии значительной готовности, сэкономить можно лишь на тестировании, что может привести к провалу проекта. Остаётся два главных фактора: время разработки и качество. Чтобы завершить проект вовремя, надо чем-то пожертвовать, но именно эти

параметры и служат критериями успеха. Так перспективный проект становится неудачным. Часто компании берутся за проекты, связанные со значительными рисками, не учитывая репутационные потери, более того, руководители не проводят расчёты для оценки условий выполнимости проекта. В результате имеет место перерасход ресурсов, чтобы выполнить проект к заданному сроку, продукт проекта получается низкого качества, заказчик не удовлетворён. Корень зла кроется в неверной оценке длительности проектного цикла. Предложенный в работе метод позволяет улучшить управление ПС за счёт строгой алгоритмизации и изменения системы оценок экспертами в процессе принятия решений о разработке проекта.

Целью данной диссертационной работы является создание системы поддержки принятия решений (СППР) для прогнозирования цикла разработки ПС на основе предлагаемого в работе метода динамической оценки (ДО).

Поставленная цель достигается в результате решения следующих основных задач:

- ✓ системного анализа опыта в области проектирования и разработки ПС;
- ✓ сравнительного анализа и классификации существующих методов оценки разработки программ и программных систем;
- ✓ разработки нового метода динамической оценки временных показателей с учётом достоинств и недостатков существующих методов;
- ✓ создание алгоритмов и методики оценки временных показателей разработки программных систем;
- ✓ апробации полученных результатов в реальных проектах;
- ✓ создание базы экспертных оценок.

Теоретической основой исследования послужили работы отечественных учёных и зарубежных авторов, специалистов в области

разработки программного обеспечения и систем поддержки принятия решений.

Наиболее значимые результаты исследований отражены в работах: DeMarco T., Lister T., Boehm B.W., Putnam L., Brooks F.P., Parkinson S.N., Fenton N.E., Pfleeger S.L., Shepperd M., Clark C. E., Архипенкова С.Я., Горбунова-Посадова М.М., Волобая А.Г., Зухба Р.Д., Куракина П.В., Малинецкого Г.Г., Махова С. А., Митина Н.А., Сорокина А.П., Торопыгина С.А. и другие.

Научная новизна полученных результатов состоит в следующем:

- ✓ предложен метод динамической оценки, позволяющий существенно увеличить достоверность расчётов времени выполнения программных проектов;
- ✓ разработан алгоритм динамического оценивания программных систем (ДОПС), реализующий предложенный метод;
- ✓ с использованием нового алгоритма разработана методика формирования базы экспертных оценок, которая позволяет на основе выполненных проектов получить числовые характеристики временных оценок группы экспертов, с сохранением их в аналитической базе данных, которая в дальнейшем используется для прогнозирования сроков завершения проектов данного класса.

Практическая значимость.

- ✓ реализован метод динамической оценки сроков завершения ИТ-проектов, дающий 10% - 15%-ный эффект в организациях с высоким уровнем зрелости для крупных ИТ-проектов, в частности, использующих методологию RUP;
- ✓ получена аналитическая база экспертных оценок, послужившая основой “виртуального эксперта”: аналитический алгоритм, который позволяет

многократно использовать оценки экспертов для прогнозирования времени выполнения проектов, имеющих схожие характеристики.

✓ разработана СППР, реализующая предложенный алгоритм;

Достоверность результатов исследования обеспечивается строгостью постановок и доказательств утверждений, корректным использованием математических моделей и стандартов, проверкой теоретических результатов реальной практической деятельностью.

Общая постановка задачи

Общая постановка задачи может быть сформулирована следующим образом - метод динамической оценки должен:

- 1) давать корректные оценки;
- 2) требовать минимальное количество экспертов для оценки;
- 3) быть применимым на практике;
- 4) быть простым и прозрачным для руководства и людей принимающих финансовые решения;
- 5) не зависеть от объёма программных проектов и выбранных методологий управления проектами;
- 6) позволять достичь скорости и точности оценки, достаточной для оперативного управления;
- 7) реализация математической модели должна быть широкодоступна и не требовательна к установке и использованию;
- 8) опираться на лучший мировой опыт, учитывать достоинства и недостатки существующих моделей и методов;
- 9) иметь возможность гибкого учёта экспертных оценок;
- 10) учитывать вероятностный характер оценок экспертов.

Глава 1. Анализ успешности проектов

1.1 Определение проекта

Дословный перевод термина “projectus” с латыни означает “брошенный вперёд”. В постоянной жизни мы постоянно осуществляем различные проекты: написание диссертации или книги, строительство дачи, подготовка ко дню рождения, ремонт, проведение исследований, и т.д. Рассел Арчибальд, классик в области управления проектами, даёт следующие определение: “Проект – это комплекс усилий, предпринимаемых с целью получения конкретных уникальных результатов в рамках отведённого времени и в пределах утверждённого бюджета, который выделяется на оплату ресурсов, используемых или потребляемых в ходе проекта”[1]. “Проект – это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов” [2,3,4], согласно определению 4 - го издания Project Management Body of Knowledge (PMBOK).

Определение проекта так же многолико как количество людей на планете, каждый из них будет понимать по-своему данное понятие, но во всех этих определениях можно выделить ряд общих черт. Может показаться, что проведения корпоративного праздника и создание новой операционной системы имеют мало общего между собой, однако, это не так. Данные виды деятельности имеют ряд схожих критериев, которые делают их проектами:

- ✓ направлены на достижение определённых целей;
- ✓ включают последовательное выполнение логически взаимосвязанных операций;
- ✓ имеют определённую ограниченную протяжённость во времени, с фиксированным началом и концом;
- ✓ неповторимы и уникальны.

Описанные выше основные характеристики отличают проекты от других видов деятельности.

1.2 Управление проектом

В начале 2000-х годов управление проектом определялось как последовательное приложение инструментов, методов, знаний и навыков к операциям проекта для удовлетворения предъявляемых к нему требований. Управление проектами выполняется посредством интеграции процессов: инициации, планирования и исполнения работ, мониторинга и контроля, управления и завершения.

Менеджер проекта (руководитель проекта) – член проектной команды, ответственный за выполнение ключевых показателей эффективности проекта - KPI (Key Performance Indicators).

В управление проектом входит:

- ✓ определение основных требований заказчика;
- ✓ установка определённых и достижимых целей;
- ✓ согласование противоречащих требований по времени, качеству, содержанию и стоимости;
- ✓ корректировка, в соответствии с ожиданиями участников проекта, планов и подходов к реализации.

Говорят о "тройственном ограничении" – содержании, времени и стоимости проекта, которые необходимо учитывать при согласовании различных требований проекта” [5,6]. Тройственная ограниченность описывает баланс между содержанием проекта, стоимостью и временем. В начале двухтысячных годов, в связи с ростом вычислительных мощностей, наблюдалась тенденция расточительного отношения к ресурсам, однако после кризисного 2008 года образовался ощутимый дефицит. В дальнейшем определение было несущественно расширено, к ограничениям проекта добавили качество и риски [7] (рис. 1).

С выходом четвертой и пятой редакции стандарта PMBOK [8] список ограничений проекта был переработан и определение управления проектами

ещё раз переписано: “Управление проектом –приложение навыков, знаний, методов и инструментов к работам проекта для удовлетворения требований заказчика, предъявляемых к проекту. Управление проектом выполняется с помощью интеграции и применения сорока двух логически сгруппированных процессов управления проектами, объединённых в пять групп процессов. Данные пять групп начинаются от инициации и планирования далее переходят в исполнение и мониторинг, поддерживаются постоянным управлением и в итоге завершаться. “В управление проектами, как правило, входит: определение требований; удовлетворение различных потребностей заказчика, в ходе планирования и выполнения проекта решение проблем и удовлетворение ожиданий заинтересованных сторон; уравнивание конкурирующих ограничений проекта” [15]:

- ✓ ресурсы;
- ✓ расписание;
- ✓ содержание;
- ✓ бюджет;
- ✓ качество;
- ✓ риски.

Процесс управления проектом проще всего представить в виде диаграммы бизнес-процессов (рис. 2).



Рис. 1. Ограничения проекта

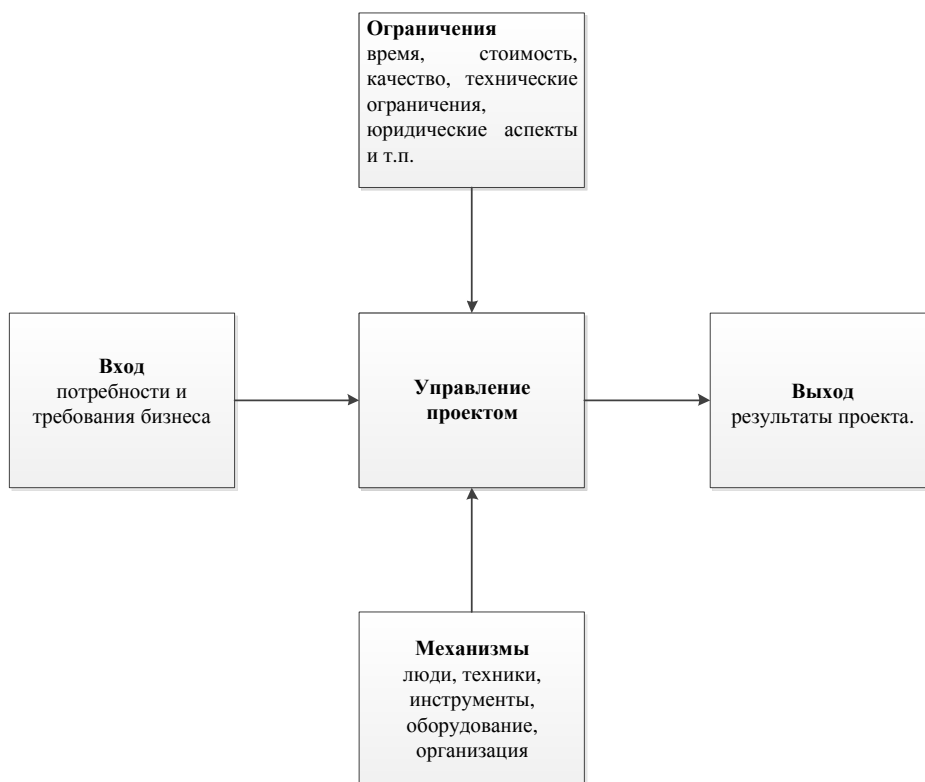


Рис. 2. Процесс управления проектом

1.3 Стандарты в области управления проектами

В определении проекта часто используются понятия неповторяемости целей, уникальности, условий реализации и результатов проектов. Подводя итог можно сделать вывод, что процесс создания новых уникальных продуктов, услуг или результатов в настоящее время стандартизован.

“Стандарты в области управления проектами нужны для: концентрации лучшей практики; взаимодействия; сертификации; системной картины. Их можно поделить на национальные стандарты (NASA PM, APMBоK, V-Modell, P2M, C-PMBoK) и стандарты с расширенной географией (ISO 10006:2003, PMBoK, PRINCE2, MSF, AIM). Существующие международные стандарты представляют собой набор рекомендаций (PMI PMBoK Guide), из которых компания самостоятельно должна формировать последовательность действий, либо тяжелы в применении за счёт большого количества обязательных процедур (PRINCE2). Другие стандарты представляют собой

требования к квалификации руководителей проектов (ICB IPMA) или недостаточно конкретны (ISO 10006). Прочие стандарты имеют существенную национальную и отраслевую специфику (P2M, ISO/IEC 12207 и др.)” [15].

Проведённые сравнения показывают, что существенная доля стандартов основана на стандарте PMBoK [2,8,11]. Стандарт PMBoK разработан институтом управления проектами (PMI), и признан стандартом де-факто. Стандарт ISO 10006:2003 [9], придал ряду положений PMBoK статус стандарта де-юре. Суть перехода от стандартов PMBOK и ISO 10006 (являющихся по своей сути рамочными стандартами), к стандарту предприятия состоит в их детализации и специализации. В то же время, несмотря на наличие готовых стандартов, существует практика создания собственных методик и руководств по управлению проектами (Oracle, IBM, Siemens и др.).

1.4 Российские национальные стандарты в области управления проектами

В 2012 году российское федеральное агентство по техническому регулированию и метрологии утвердило национальные стандарты в области управления проектами - ГОСТ Р 54869, 54870, 54871. Описывающих требования к управлению проектом, программой проектов и портфелем проектов [89].

Утверждение российских стандартов по управлению проектами - прогрессивное решение. До этого Россия не имела стандартов и рекомендаций, содержащих требования к реализации программных проектов. Российские компании использовали иностранные методические документы. Во времена СССР существовала методологическая база, позволяющая реализовывать масштабные проекты, но даже тогда универсальных требований к процедурам и процессам управления проектов создано не было.

1.5 Риски проектов и первичная оценка

“Проект без риска — удел неудачников. Риски и выгода всегда ходят рука об руку”. [10] Риск проекта – это неопределённое возможное событие или условие, которое в случае возникновения может иметь как позитивное так и негативное воздействие, на один из КРІ проекта. Риск может быть вызван несколькими комбинированными причинами и в случае возникновения, чаще всего, оказывать негативное воздействие на несколько КРІ. [2,8,11] Вероятность возникновения риска – вероятность того, что риск наступит. Риск с нулевой вероятностью не считается риском, риск с вероятностью 100% является гарантированным событием, это факт, который необходимо учитывать в ходе планирования ресурсов. Риск может иметь как негативное, так и положительное влияние на проект. Положительный риск называют “шансом”.

Управление рисками. “Мы не боремся с рисками — мы ими управляем” [12]. Управление рисками включает в себя процессы, относящиеся к планированию, анализу, идентификации и реагированию, мониторингу и управлению. Данные процессы подлежат обновлению в ходе всего цикла проекта. Цель управления рисками – увеличение вероятности возникновения и степени воздействия благоприятных событий и снижение вероятности возникновения и степени воздействия неблагоприятных событий.

Процессы управления рисками проекта включают:

- ✓ начальную идентификацию рисков;
- ✓ подготовку и планирование управления ими;
- ✓ количественный анализ рисков;
- ✓ качественный анализ рисков;

- ✓ выработка плана реагирования на риски;
- ✓ управление и постоянный мониторинг рисков.

1.6 Основные причины провала ИТ - проектов:

- ✓ отсутствие требований заказчика;
- ✓ неполнота требований и их частое изменение;
- ✓ отсутствие требуемого опыта и ресурсов;
- ✓ “забытые работы” и неполнота планирования;
- ✓ отсутствие взаимодействия с заказчиком;
- ✓ грубые ошибки в оценке трудоёмкости, сроков и длительности работ [13].

Примерами областей, которым сопутствует значительный риск, могут служить:

- ✓ качество и стабильность требований пользователя;
- ✓ стабильность и полнота описания внешних интерфейсов;
- ✓ опыт и квалификация кадров;
- ✓ техническая новизна проекта”. [14]

Особенно важен анализ риска при планировании работ, одним из главных рисков при разработке ПО является риск возникновения ошибок при **оценке трудоёмкости и сроков проекта.**

1.7 Анализ успешности проектов

Анализ результатов проектной деятельности в области ИТ показал существующую тенденцию к снижению качества ИТ-проектов. Основной причиной ухудшения качества служит “выбывание квалифицированных руководителей проектов, значительное увеличение объёма работ и, как следствие, усложнение процессов управления и интеграции” [15].

Todd Little, из Landmark Graphics в статье “Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty” пишет: “Software development project schedule estimation has long been a difficult problem. The Standish CHAOS Report indicates that only 20 percent of projects finish on time relative to their original plan. 1. Conventional wisdom proposes that estimation gets better as a project progresses. This concept is sometimes called the cone of uncertainty, a term popularized by Steve McConnell. 2 .The idea that uncertainty decreases significantly as one obtains new knowledge seems intuitive project estimation”.

[16]

На протяжении многих лет исследователи и практики пытаются анализировать, как добиться успешного управления в области ИТ-проектов. В числе них Standish Group (SG), которая регулярно публикует свои исследования в Chaos Report [38,39] (CR) (табл. 1). CR компании SG является основным индикатором проблем в области разработки программного обеспечения.

Таблица 1. The Chaos Report (данные приведены в процентах)

Год	Успешные проекты	Неудачные проекты	Провалившиеся проекты
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2008	32	44	24
2010	37	42	21
2012	39	43	18

Согласно данным за 2006 докризисный год - 65 % современной IT-индустрии работало впустую. Подход SG имеет существенные проблемы: обманчивые и односторонние определения, искажение практической оценки и, как следствие этого, – бессмысленные данные.

Первая проблема CR - обманчивые определения. SG разбила проекты на три категории: успешные проекты, неудачные проекты и провалившиеся проекты. Успешные проекты – проект завершён вовремя, в рамках бюджета с достижением всех заявленных первоначально свойств и функций. Неудачные проекты – проект завершён и работает, но с превышением бюджета или времени на разработку, и с меньшим количеством заявленных первоначально функций и свойств. Провалившиеся проекты – проект отменен в течение цикла разработки. Введём отношение прогнозируемой величины к актуальной – $K = F / A$; (Forecast/Actual), тогда K_C – отношение по стоимости; K_T – отношение по времени; K_F – отношение по величине функциональности. Тогда успешные проекты – проект завершён $K_C \geq 1, K_T \geq 1, K_F \leq 1$; неудачные проекты – проект завершён и работает $K_C < 1, K_T < 1, K_F > 1$.

Вывод: определение успешности проекта обманчиво, так как оно базируется исключительно на правильности первоначальной оценки.

Вторая проблема CR - односторонние определения. Определения SG пренебрегают работой с недогрузкой по времени и цене, и перегрузкой по функциональности. Возьмём данные крупной компании А (реальные данные, финансовый сектор), построим график K_C относительно процента завершения проекта (рис. 3), прогнозируемая стоимость находится около актуального значения 1. Фактор оценки качества первоначального прогноза (Estimating Quality Factor - EQF) (см. главу 3.14) EQF=8,5. Половина проектов имеют отклонение по времени около 12% и ниже. Согласно определению SG результат успешности равен 59% .

Вывод: организация Y получает низкий уровень успеха в прогнозировании цены, благодаря односторонности определения показателя.

Третья проблема CR - искажение практической оценки. Следование стандартным определениям служит причиной крупных временных и стоимостных преувеличений, является основой занижения функциональности. Для компании В (реальные данные, ИТ - проекты) отношение K_c (рис. 4) значительно превышает 1. Все проекты имеют избыточный бюджет, так как руководители проектов преувеличивают бюджет для увеличения запаса надёжности. $EQF=0,43$ более половины проектов имеет отклонение по стоимости порядка 233%.

Четвертая проблема CR - бессмысленные данные. Для компании С (реальные данные, добыча полезных ископаемых) (рис. 5) большинство предварительных оценок ниже актуальных, проекты шли дольше, чем предварительно ожидалось. $EQF=4,7$, половина проектов имеют отклонение по времени 21% или менее.

Огромное влияние на практический результат оказывает завышение первоначальные оценки по стоимости или занижение по времени.

Принятие стандартных определений. Проведём стандартную оценку согласно определению SG (см. табл. 2). Согласно данной оценке - Компания А получается успешной, С является аутсайдером, В находится в середине.

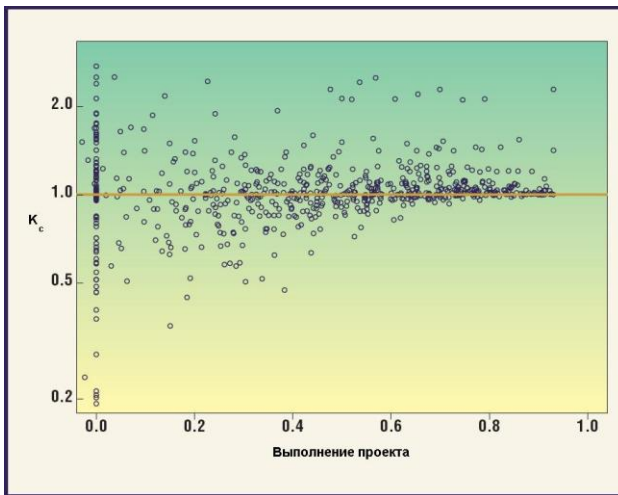


Рис. 3. Односторонние определения

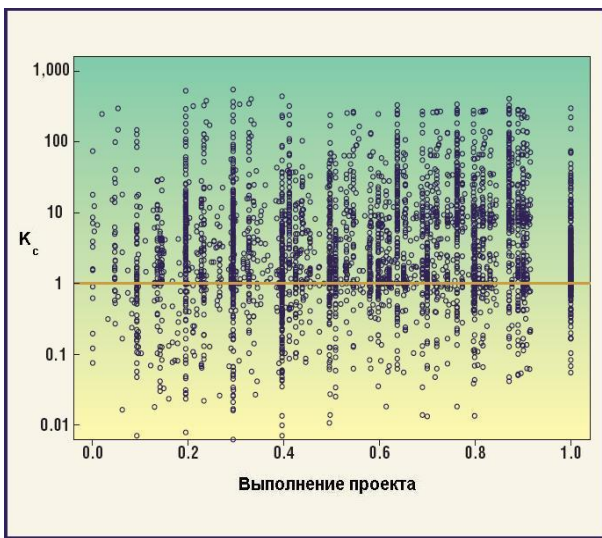


Рис. 4. Искажение практической оценки

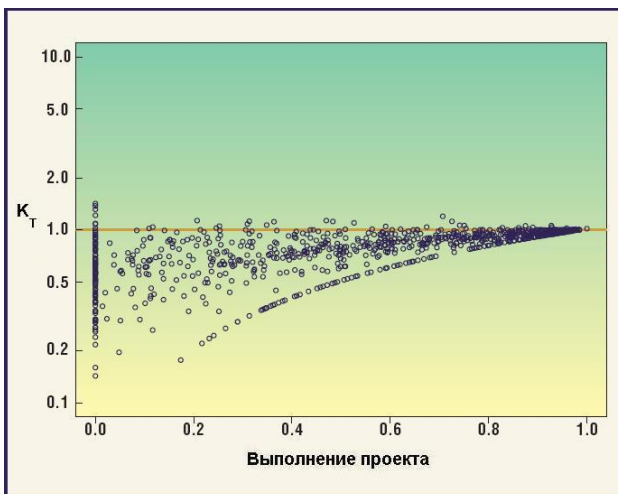


Рис. 5. Бессмысленные данные

Таблица 2. Оценка успешности проекта, согласно определению SG
(проценты)

Компания	Успешные проекты	Неудачные проекты	(EQF)
Компания А	67	33	1.1
Компания В	5.8	94.2	2.3
Компания С	59	41	6.4
1- С	94.2	5.8	2.3

Компания С превращается из аутсайдера в лидеры с 94% успешных проектов в том случае, если необходимо оценивать не минимальное время, необходимое на проект, а максимальное.

Вывод: Применение стандартах правил SG для организации не отражают реальности, большое влияние оказывают первоначальные оценки, агрегация большого количества этих данных не реалистична и бессмысленна.

1.8 Анализ успешности проектов в России

По данным за 2013 год опрос PM Expert 100 отечественных компаний, позволил выявить основные проблемы российских организаций. К ним относятся: отсутствие актуальной картины проекта; задержки в ходе выполнения; отсутствие чётко распределённой ответственности. После внедрения системы управления, улучшение ситуации отметили 30% респондентов, 65% указали, что ситуация значительно улучшилась. Использование стандартных методологий управления проектами показывает, что оценки российских заказчиков вполне совпадают с мировой статистикой. Данные, представленные в аналитическом исследовании компании IBM, говорят о том, что прирост стоимости по результатам управляемого проекта достигает 20% по сравнению с тем, как если бы он выполнялся без применения соответствующей методологии. [3]

1.9 Анализ основных методов и метрик разработки программных проектов

1.9.1 Методы, основанные на принципах

Price-to-win. Основа метода - независимо от планируемых затрат на разработку проекта, оценка стоимости ПО изменяется в соответствии с требованиями заказчика. Price-to-win есть политика проведения переговоров с клиентом, поэтому применяется в компаниях, не имеющими ресурсов для качественной оценки проектов. Использование данного метода влечёт для руководителя проекта ряд негативных последствий: невыполнение сроков сдачи, нехватку проектных ресурсов, как результат – потеря клиента.

Оценка по Паркинсону. Основа метода поддержка принципа: “Объем работы возрастает в той мере, в какой это необходимо, чтобы занять время, выделенное на её выполнение” [20]. Данный метод был высказан Parkinson S.N. и описывает взаимоотношения в административных институтах, показывая неэффективное использование ресурсов [20]. В части разработки ИТ - проектов, метод Parkinson S.N. используется в виде следующей парадигмы – *“для повышения производительности труда разработчика, необходимо уменьшить время, отведённое на разработку”* [19]. Использование принципа несёт перегрузку и дисбаланс ресурсов из-за этого увольнение ключевых сотрудников и как следствие срыв сроков выполнения проекта.

1.9.2 Линейный подход

Стоимость разработки ПО (C) - линейное произведение трудозатрат (WH) на их удельную стоимость (CPU) (табл.3).

$$COST = WH \times CPU \quad (F.1)$$

Таблица 3. Обозначения и сокращения, используемые при линейном подходе

	Обозначение	Сокращение
Стоимость разработки ПО	COST	C
Трудозатраты	Working Hours	WH
Удельную стоимость	Cost Per Unit	CPU
Размер исходного кода	Lines Of Code	LOC
Временная производительность	Productivity	P

CPU при для разработки ПО формируется из ФОТ (фонд оплаты труда) и связанных начислений сотрудников, иные составляющие имеют меньший порядок, и ими часто пренебрегают. Расчёт WH и C, при разработке ПО, основан на следующем подходе:

$$WH = LOC \times P \quad (\text{F.2})$$

$$COST = LOC \times P \times CPU \quad (\text{F.3})$$

Исходя из (F.3) - стоимость разработки ПО линейно зависит от LOC, т.е. количества строк написанного кода. Согласно мнению Brooks F.P, “наши методы оценки ошибочно путают достигнутый прогресс с затраченными усилиями”[21]. Данное утверждение относится к методу измерения результата. В начале прошлого века, не удалось найти ничего лучшего, чем использование для расчёта LOC. При эволюции программиста от студента к специалисту происходит рост мастерства, код для решения схожих задач становится компактнее. При данном подходе стоимость разработки линейно зависит от LOC, данный метод оценки стимулирует писать большое количество некачественного кода.

1.9.3 Software Life-cycle Model (SLIM)

Наиболее распространённая модель аналитической группы - модель SLIM (предложена Putnam L. в 1978 г.), основана на основном утверждении,

что основные затраты на разработку ПО распределяются согласно кривым Putnam–Norden–Rayleigh (кривая PNR), являющиеся графиками функции распределение рабочей силы по времени [22].

После ряда эмпирических наблюдений, Putnam L. выразил аналитическое уравнение модели в виде:

$$S = E^{1/3} \cdot t^{4/3} \cdot P \cdot B^{1/3} \quad (\text{F.4}), \text{ где}$$

Таблица 4. Обозначения и сокращения, используемые в модели SLIM

Сокращение	Обозначение	Расшифровка										
S	Size	Размер проекта в ESLOC										
ESLOC	Effective Source Lines of Code	Эффективное количество строк исходного кода (или количество функциональных точек) $ESLOC=K*LOC$ <table border="1" data-bbox="774 1008 1420 1254" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Формат LOC</th> <th>K</th> </tr> </thead> <tbody> <tr> <td>Новый</td> <td>1</td> </tr> <tr> <td>Повторно используемый</td> <td>0,55</td> </tr> <tr> <td>Автоматизированный</td> <td>0,40</td> </tr> <tr> <td>Входные данные для инструментов</td> <td>0,5</td> </tr> </tbody> </table>	Формат LOC	K	Новый	1	Повторно используемый	0,55	Автоматизированный	0,40	Входные данные для инструментов	0,5
Формат LOC	K											
Новый	1											
Повторно используемый	0,55											
Автоматизированный	0,40											
Входные данные для инструментов	0,5											
B=B(S)	Scaling factor	Масштабирующий коэффициент, зависит от размера проекта										
P	Productivity	Производительность										
E	Effort	Объем работы в человеко-годах										
t	Time	Общее время разработки в годах										

Объем работы в человеко-годах, получаем зависимость от размера строк кода:

$$E = S^3 \cdot P^{-3} \cdot B \cdot t^{-4} \quad (\text{F.5})$$

Для повышения качества прогноза SLIM позволяет вести калибровку на основе хронологической информации. Модель не приобрела широкой популярности распространение в силу своей специфичности (применима

только для крупных проектов), однако существуют организации и проектные команды, успешно использующие ее и сегодня. К недостаткам модели можно отнести все недостатки линейного подхода. В случае использования модели SLIM зависимость основных показателей от LOC снять не удалось.

1.9.4 Функциональные точки (Function Points)

Функциональные точки (FP) стали заменой LOC для измерения затрат на разработку ПО (предложены Albrecht A. в 1979 г). В середине 70-х возникла необходимость в отличном от LOC подходе к оценке трудозатрат на разработку ПО. Создатель метода FP хотел найти новый подход, независимый от среды разработки и языка программирования. С 1986 г. разработку стандарта для FP поддерживает International Function Point User Group (IFPUG) [23], которая разрабатывает руководство по применению и расчёту FP - Function Point Counting Practices Manual (FPCPM). FPCPM версии 4.1 признана ISO в качестве стандарта оценки размера ПО [24]. Методика FP основана на разграничении взаимодействия компонент разрабатываемой системы (рис. 6). Система делится на классы компонент по формату и типу логических операций. В основе деления лежит следующие правило - область взаимодействия программы делится на взаимодействие компонент внутри приложения и взаимодействие с иными приложениями.

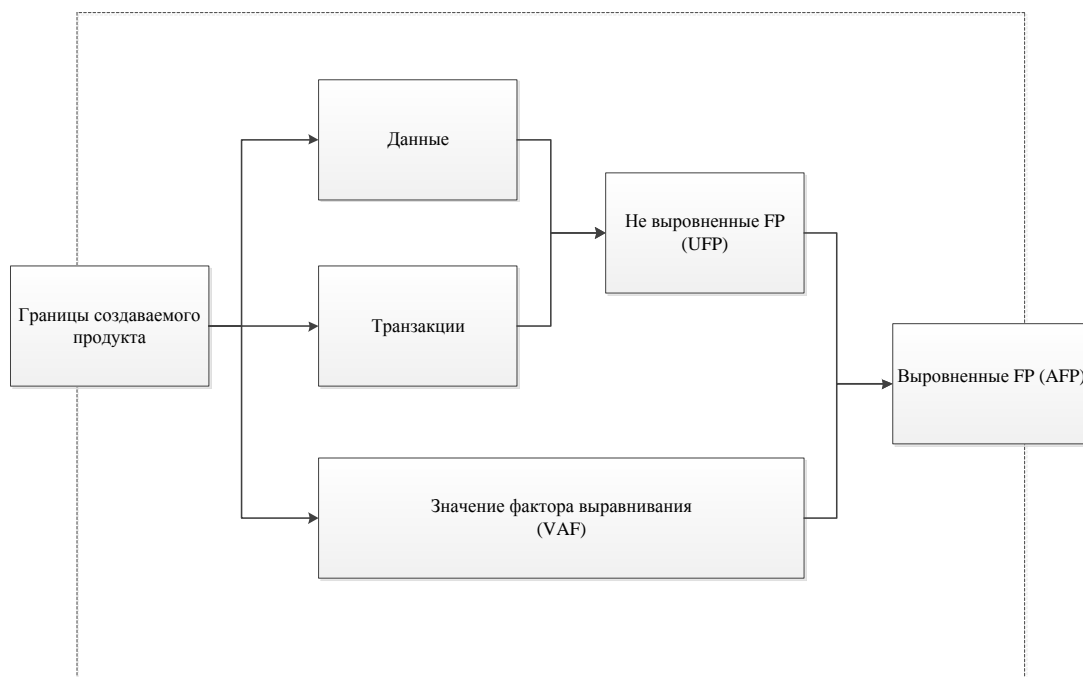


Рис 6. Процедура оценки по методу функциональных точек.

Транзакция (Тр) – элементарный замкнутый и неделимый процесс, имеющий значение для пользователя, переводящий систему из одного состояния в другое (табл. 9). В соответствии с FPCPM [24] используется пять классов компонентов, на основе которых проводится анализ (см. табл. 5):

Таблица 5. Классы компонент в соответствии с FPCPM

Класс компонент	Сокращение	Описание
Внешний запрос	External Inquiry (EQ)	Тр в результате которой происходит ввод\вывод.
Внешний вывод	External Output (EO)	Тр в результате которой данные выходят за границу приложения изнутри.
Внешний ввод	External Input (EI)	Тр в результате которой данные выходят за границу приложения.
Внешний интерфейсный файл	External Interface File (EIF)	Данных вне приложения и являющихся ILF для другого приложения.
Внутренний логический файл	Internal Logical File (ILF)	Данные внутри приложения получаемые внешним вводом.

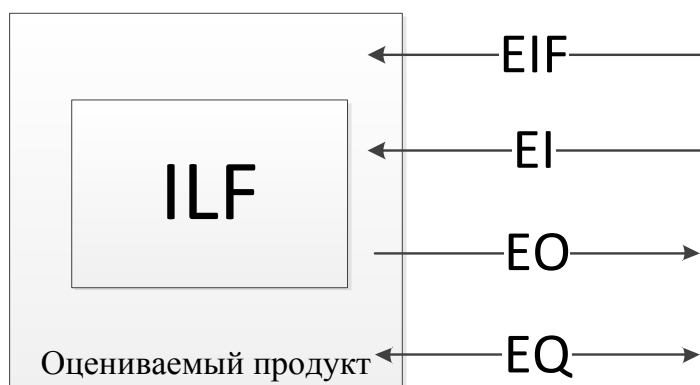


Рис 7. Границы транзакций

Все компоненты приложения делятся по сложности и относятся к трем категориям (низкого, среднего, высокого) уровня. Для EI, EO, EQ сложность определяется по числу элементов определённых данных - Data Element Types (DET) и числу транзакционных файлов - File Types Referenced (FTR). DET это уникальное, неповторяемое поле данных, например, номер счета клиента в банке - 1 DET, индекс, город, улица – 3 DET. Для файлов ILF и EIF значимы DET и Record Element Types (RET), RET есть тип элемента записи.

Для уровней сложности в FPCPM [24] введено соотношение между FTR и DET (табл. 6). На основании данных соотношений проходит оценка в не выровненных функциональных точках Unaligned FP (UFP) по трем градациям: LOW(L), AVERAGE(A), HIGH(H). (табл. 7,8).

Таблица 6. соотношение между FTR и DET.

	До 19 DET	20-50 DET	Более 50 DET
1 RET	L	L	A
2-5 RET	L	A	H
Более 6 RET	A	H	H

Таблица 7. Оценка сложности ILF в не выровненных UFP

Сложность данных ILF	Количество не выровненных UFP
L	7
A	10
H	15

Таблица 8. Оценка сложности EIF в не выровненных UFP

Сложность данных EIF	Количество не выровненных UFP
L	5
A	7
H	10

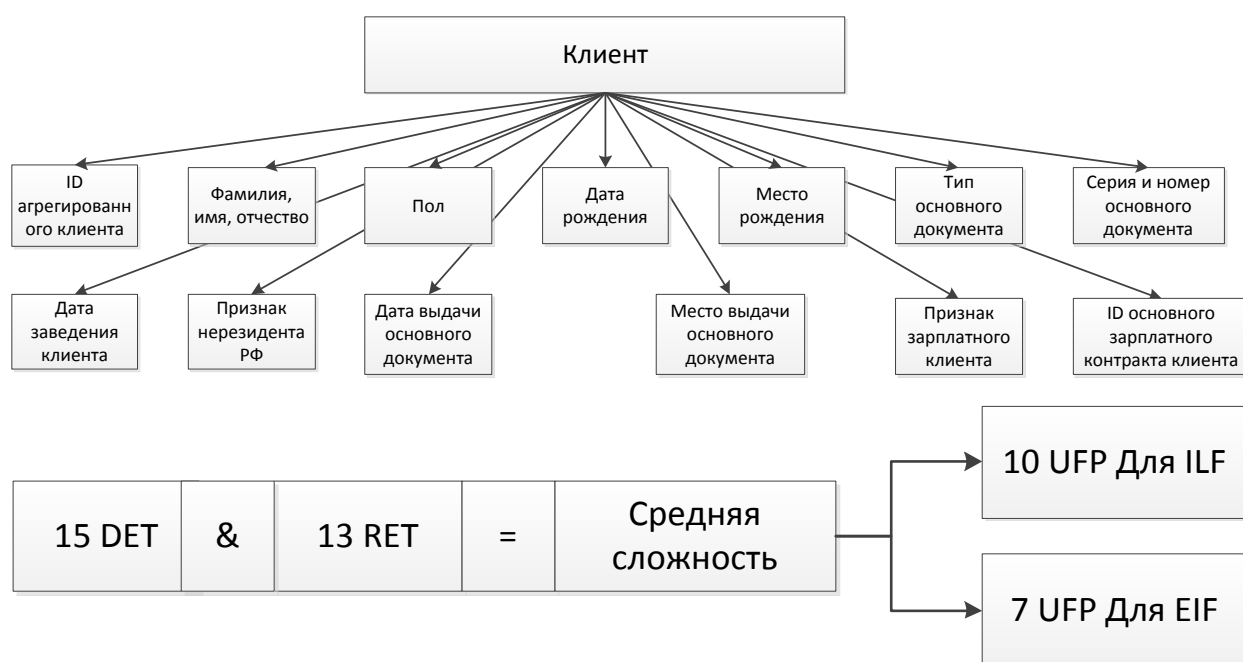


Рис 8. Оценка данных "Клиент Банка"

Таблица 9. Виды транзакций

Функция	Тип транзакции		
	ЕІ	ЕО	ЕQ
Изменение поведения системы	Основная	Дополнительная	Не применима
Поддержка одного и \ или более ILF	Основная	Дополнительная	Не применима
Вывод информации пользователю	Дополнительная	Основная	Основная

File Type Referenced (FTR) - даёт возможность рассчитать количество объектов ILF и EIF считываемых и \ или модифицируемых в одной транзакции (табл. 10,11,12,13).

Таблица 10. Характеристики сложности транзакции ЕІ

ЕІ	До 4 DET	5-15 DET	Более 16 DET
До 1 FTR	L	L	A
2 FTR	L	A	H
Более 3 FTR	A	H	H

Таблица 11. Характеристики сложности ЕО,ЕQ

ЕО & EQ	До 5 DET	6-19 DET	Более 20 DET
До 1 FTR	L	L	A
2- 3FTR	L	A	H
Более 4 FTR	A	H	H

Таблица 12. Сложность EI, EQ

Сложность транзакций EI, EQ	Количество не выровненных UFP
L	3
A	4
H	6

Таблица 13. Сложность EO

Сложность транзакций EO	Количество не выровненных UFP
L	4
A	5
H	7

Далее происходит расчет объема продукта в UFP:

$$UFP = \sum_{i=1}^{n_1} UFP_{ILF,i} + \sum_{j=1}^{n_2} UFP_{EIF,j} + \sum_{k=1}^{n_3} UFP_{EI,k} + \sum_{m=1}^{n_4} UFP_{EO,m} + \sum_{f=1}^{n_5} UFP_{EQ,f} \quad (\mathbf{F.6})$$

Фактор выравнивания определяется следующим образом

$$VAF = (TDI \times 0,001) + 0,65 \quad (\mathbf{F.7}),$$

$$TDI = \sum_{i=1}^n DI_i \quad (\mathbf{F.8}), \text{ где}$$

Total degree of influence (TDI) определяется как расчет кумулятивного эффекта системных характеристик - Degree of Influence (DI) (табл. 14), оцениваемых с шагом один от нуля до 5.

Таблица 14. Системные характеристики

№	Системные характеристики
1	Обмен данными
2	Распределенная обработка данных.
3	Производительность.

4	Ограничения по аппаратным ресурсам.
5	Транзакционная нагрузка.
6	Интенсивность взаимодействия с пользователем.
7	Эргономика.
8	Интенсивность изменения данных ILF пользователем.
9	Сложность обработки.
10	Повторное использование.
11	Удобство инсталляции.
12	Удобство администрирования
13	Портируемость
14	Гибкость

Расчет количества выровненных FP для проектов разработки – Development Functional Point (DFP) и развития – Enhancement Functional Point (EFP) различен:

$$DFP = (UFP + CFP) \times VAF \quad (\mathbf{F.9})$$

$$EFP = (ADD + CHGA + CFP) \times VAFA + (DEL \times VAFB) \quad (\mathbf{F.10})$$

Где,

VAFA фактор выравнивания, рассчитанный после завершения проекта.

VAFB значение фактора выравнивания до старта проекта.

ADD FP для добавления функциональности.

CFP дополнительные FP, требуемые при установке.

CHGA FP для измененных функций, рассчитанные после модификации.

DEL объем удаленной функциональности.

Достоинства и недостатки FP:

1) измерения не зависят от применяемой платформы;

- 2) метод обеспечивает единообразный подход к оценке всех проектов организации;
- 3) применение FP основано на анализе требований, в таком случае оценка трудозатрат может быть выполнена на ранних стадиях проекта;
- 4) оценки продолжают уточняться по ходу жизненного цикла
- 5) высокая сложность и трудоёмкость метода;
- 6) большая потеря времени на повторный анализ. На начальном этапе проектирования часто не возможно понять количество предполагаемых объектов, транзакций и операций ввода \ вывода. По мере продвижения разработки придется проводить повторный анализ и пересчитывать основные показатели.

1.9.5 Точки свойств

В условиях, когда сформулированные требования к системе не отражают сложности реализации (управление процессом в реальном времени, системные приложения, и т.д.), метод FP себя не оправдывает. Для решения задач расчёта размера указанного ПО Capers J. в 1986 предложил экспериментальную методику анализа характеристических точек - Feature Points [25], которая позволяет проводить учет не только требования к системе, но и внутренних особенностей ее реализации. Основа метода состоит в оценке количества алгоритмов в исходной программе и модифицирует степень значимости для расчёта FP.

1.9.6 Метод Mark II

Модификация метода FP, представлена Symons C. в 1988 г. Автор стремился избавиться от известных недостатков и сделать метод FP пригодным для оценки сложных систем. Метод позволяет получить схожие результаты как при оценке системы целиком, так и при сложении оценок, полученных для составляющих подсистем.

1.9.7 3-х мерные и объектные FP

Развитие метода FP подразделением корпорации Boeing, занимающейся разработкой ПО, предложено в 1991 г. Основа метода - сложность любой задачи в программной среде можно представить в трех различных измерениях – сложность вычислений, количество вводов/выводов и управляющая логика. Классическая постановка задачи метода FP не предусматривает использование объектно-ориентированного подхода. В настоящее время, в проектах используется адаптированный вариант метода, использующий терминологию объектно-ориентированного подхода

Несмотря на существенные достижения рассмотренных выше методов оценки параметров реализации программных проектов, опытный руководитель проекта скажет – “лучший способ узнать длину пути – это пройти его”[26]. Описанные ниже методы оценивания основаны на анализе эмпирических данных

1.9.8 Метод оценки PERT

В 50-е годы две независимые проектные команды представили методы управления сложными комплексами работ[27,28]. Компании Remington Rand и Du Pont предложили “метод критического пути (Critical Path Method(CPM)) [29]. Параллельно в армии США в рамках проекта Polaris был создан метод для анализа и оценки длительности реализации работ проекта - Program Evaluation & Review Technique (PERT). Разработчики PERT: проектное бюро BMC; Booz; Lockheed Air Craft; Allen & Hamilton. Благодаря PERT проект Polaris удалось закончить на два года раньше. Успешное завершение проекта Polaris способствовало популяризации данного метода.

Методы CPM и PERT используют метод сетевых диаграмм, CPM оперирует единой длительностью работы, в отличие от PERT, который учитывает четыре: наиболее вероятную, средневзвешенную,

оптимистическую и пессимистическую. Выбранный подход обусловлен различными сферами применения методов. Метод PERT “родился” при реализации проекта с высокой степенью неопределенности, неопределенности проектной среды СРМ была невысока, и разработчики могли дать более точные оценки длительности работ, опираясь на свой предыдущий опыт. За прошедшее время произошла взаимная интеграция методов. PERT полностью совпадает с СРМ, за исключением того, что PERT оперирует тем, что продолжительность любой операции лежит в пределах, исходящих из статистического распределения, используется три оценки для расчета времени длительности каждой операции:

- ✓ наилучшее (оптимистическое);
- ✓ наиболее вероятное (средний показатель);
- ✓ наихудшее (пессимистическое).

Разработчики PERT для расчета продолжительности одной операции выбрали аппроксимацию Бета - распределения:

$$t_e \sim B(\alpha, \beta) \quad (\text{F.11}), \text{ где Плотность распределения случайной}$$

величины t_e задаётся плотностью вероятности:

$$f_X(x) = \frac{x^{\alpha-1} \times (1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad (\text{F.12})$$

$$\alpha > 0, \beta > 0$$

$$B(\alpha, \beta) = \int_0^1 x^\alpha \times (1-x)^{\beta-1} dx \quad (\text{F.13}) - \text{Бета функция}$$

Распределение длительности программного проекта есть сумма средневзвешенных показателей операций на критическом пути (КП).

$$T = \sum_{e=1}^k t_e, \text{ (F.14)}$$

где k - количество операций на критическом пути.

Время операции определяется по следующей формуле:

$$t_e = \frac{o + p + 4m}{6} \text{ (F.15),}$$

где: t_e - средневзвешенное время одной операции e ;

p - пессимистическое время;

m - наиболее вероятное время операции.

o - оптимистическое время;

Среднее значение применяют к сети проекта, рассчитывают время завершения проектных работ и резерв по времени согласно методологии СРМ. Отклонения в оценках определяются согласно следующим уравнениям(F.16),(F.17). Уравнение (F.16) представляет стандартное отклонение для одной операции.

$$\sigma_{t_e} = \frac{p - o}{6} \quad (\text{F.16})$$

$$\sigma_T = \sqrt{\sum_{e=1}^k \sigma_{t_e}^2} \quad (\text{F.17}).$$

Формула (F.17) описывает стандартное отклонение проекта и включает в себя операции находящиеся на КП.

T_E (средняя продолжительность проекта) - сумма средних, отведенных на выполнение операций на КП проекта, которая согласно центральной предельной теореме стремится к нормальному распределению.

Исходя из средней продолжительность проекта и дисперсии операций, можно произвести расчет выполнения проекта к определенному времени. Уравнение (F.18) используют при расчете отклонений:

$$Z = \frac{T_S - T_E}{\sqrt{\sum_{e=1}^k \sigma_e^2}} \quad (\text{F.18}), \text{ где}$$

T_E - длительность КП;

T_S - продолжительность работы;

Приведем прикладной пример использования PERT (рис.9)

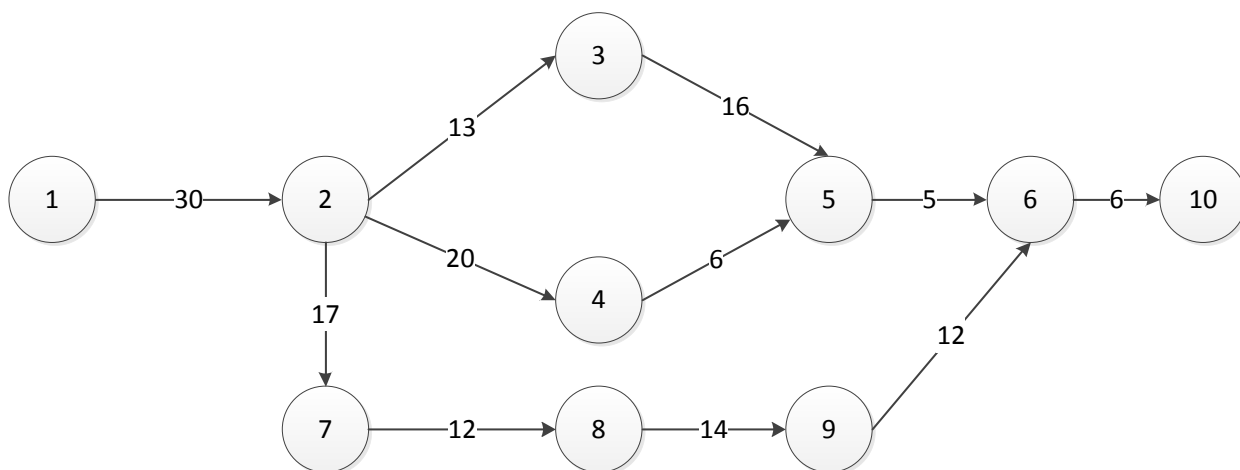


Рис. 9. Сеть прикладного пример использования метода PERT

Сеть проекта представлена на рис.9. Продолжительность операций и значение стандартного отклонения для операции представлены в таблице 15 .

Таблица 15. Характеристики сети рис. 8

Операция	o	p	m	t_e	$\sigma^2_{t_e}$
1-2	17	47	29	30	25
2-3	6	24	12	13	9
2-4	16	28	19	20	4
2-7	6	24	18	17	9
3-5	13	19	16	16	1
4-5	2	14	5	6	4
5-6	2	8	5	5	1
6-10	2	10	6	6	$\frac{16}{9} \approx 1,78$
7-8	8	16	12	12	$\frac{16}{9} \approx 1,78$
8-9	10	18	14	14	$\frac{16}{9} \approx 1,78$

9-6	8	16	12	12	$\frac{16}{9} \approx 1,78$
6-10	2	14	5	6	4

Критический путь – 1,2,7, 8, 9, 6,10. Продолжительность критического пути $T_E = 91$ единица времени. Имея эту информацию и используя статистические методы, рассчитаем вероятность выполнения проекта к моменту времени 80.

$$T_S = 77, \sigma_T \approx 6,58, Z = \frac{T_S - T_E}{\sqrt{\sum_{e=1}^k \sigma_e^2}} = \frac{91 - 77}{6,58} \approx 2,13, P \approx 0,98$$

Значение $Z \approx 2,13$ означает 98 процентную вероятность завершения проекта ранее 98 единицы времени.

Достоинства и недостатки PERT (CRM):

- 1) применение метода в новых инновационных проектах затруднено, так как данные проекты обладают большой степенью неопределенности. Экспертные оценки случайны, из-за отсутствия личного опыта эксперта;
- 2) один из самых удачных методов, зарекомендовавших себя на практике. Возьмем его за основу первого шага модели.

1.9.10 Метод Дельфи (Delphi)

Подход к экспертной оценке согласно методу Delphi, созданный в Rand Corporation в 1966 г., нашел свое применение в ИТ – проектах, вариация данного метода названа - Wideband Delphi. Основа метода - с помощью серии интервью, опросов и последовательных мозговых штурмов, добиться

максимального согласия при определении верного решения. Анализ проводится в несколько последовательных этапов, результаты этапов обрабатываются статистическими методами. Базовый принцип метода - некоторое количество независимых экспертов лучше оценивает и предсказывает результат, чем структурированная рабочая группа проекта.

Подход Wideband Delphi позволяет избежать:

- ✓ открытых столкновений между экспертами с различными мнениями (исключен непосредственный контакт экспертов);
- ✓ группового влияния, возникающего при совместной работе (принятие мнения большинства);
- ✓ личного присутствия экспертов (возможно, проводить опрос удаленно не собирая экспертов в одном месте).

Порядок проведения. Эксперт или группы экспертов отвечают индивидуально в письменной форме. Организационная группа подводит итоги, объединяя мнения экспертов. Этапы метода: предварительный; основной; аналитический (табл. 16).

Таблица 16 Этапы метода Delphi.

Этап	Описание
Предварительный	
Подбор экспертов.	Выбор экспертов для анализа.
Основной	
Постановка проблемы.	Рассылка обсуждаемого вопроса с требованием разбить его на под вопросы. Организационная группа формирует общий опросник, на основании наиболее часто

	встречающихся подвопросов.
Рассылка общего опросника.	Эксперты дают обратную связь, на основе ответов экспертов составляется улучшенный опросник (УО).
УО.	УО рассылается экспертам, которым необходимо дать вариант решения и рассмотреть наиболее радикальные точки зрения других экспертов.
Выявление преобладающих мнений.	Выявляют преобладающие мнения, происходит последовательное сближение точек зрения. Экспертов знакомят с доводами тех, чьи наиболее радикальны. Происходит переоценка мнений и повторении итерации.
Выработка окончательной оценки и практических рекомендации.	Итерации повторяются до согласованности мнений, или принятия решения об отсутствии единого мнения.
Аналитический	
Подведение итогов	Организационная группа проводит проверку согласованности мнений экспертов, анализ полученных выводов и подводит итоги.

Достоинства и недостатки:

- 1) беззащитность эксперта перед организационной группой;
- 2) не обязательно правильное мнение большинства продавливается, наиболее эффективное решение меньшинства отбрасывается;
- 3) анализ занимает длительное время;

- 4) метод применим для стратегического планирования, но не применим для оперативного анализа;
- 5) возрастает стремление экспертов попасть в группу большинства;
- 6) существует возможность манипуляции организационной группой над мнениями экспертов;
- 7) требует изложения мнения экспертов в посменной форме;
- 8) эксперты должны обладать высоким уровнем мотивации;
- 9) экспертные оценки приводят к хорошим результатам при невысоких затратах. Однако использование предыдущего опыта не подразумевает отказа от использования математических моделей. Большинство современных методов оценки с привлечением эмпирических данных использует адаптацию параметрических моделей.

1.9.11 Экспертная оценка

Данный метод основан на принципе Wideband Delphi, широко используется в инновационных проектах или процессах решающих инновационные задачи [30]. К оценке привлекаются разработчики, которые оценивают свою часть проекта. Результаты оценок экспертов собираются в единую целостную систему. Оценки экспертов заносятся в протокол и открыто обсуждаются. Опрос экспертов осуществляется по методу Wideband Delphi, достигается баланс оценки, далее идет следующая итерация, по мере прохождения повторяемых итераций точность оценки проекта существенно повышается [31]. Данный метод оценивания несет все недостатки и достоинства дельфийской методики.

1.9.12 Оценка по аналогии (ОПА)

Разновидность экспертной оценки, выделяемая в отдельный метод [32]. ОПА использует эмпирические данные характеристик завершённых проектов. Основное различие метода ОПА состоит в следующем – метод

позволяет выделить схожие проекты в отличие от алгоритмических моделей, которые используют эти данные для калибровки чувствительности параметров. Схема оценки состоит из нескольких этапов:

- ✓ сбор данных по разрабатываемому проекту;
- ✓ поиск и анализ аналогичных проектов по выбранным характеристикам разрабатываемому;
- ✓ экспертная оценка разрабатываемого проекта.

Для выбора проектов, наиболее близких оцениваемому, используется норма в n - мерном пространстве. Характеристикам проекта присваивается веса, согласно значимости. Проекты и их характеристики отображаются в n – мерном пространстве как точки, вычисляется расстояние между точками [32]:

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (\text{F.19}),$$

где a и b – точки в пространстве, a_i, b_i координаты точек в плоскостях.

У проектов, имеющих наибольшее сходство, значение формулы (F.19) будет минимальным. Данный метод оценивания несет все недостатки и достоинства дельфийской методики.

1.9.13 СОСОМО,II

Семейство моделей Constructive Cost Model (COCOMO) разработано Boehm V.W. в 1981 г. на основе проектного опыта TRW Aerospace [33]. Boehm V.W. на тот момент был уже известным практиком и теоретиком в области развитие научных подходов к управлению программными проектами (разработал спиральную модель проектирования ПО и Wideband Delphi). Модель СОСОМО разработана на основе анализа статистических данных

более 50 различных типов проектов. Модель в своей основе содержит 3 режима использования и несколько уровней детализации [31].

Таблица 17. Режимы моделей семейства COSOMO

Название режима	Размер проекта	Описание уровня детализации
Внедренный	Более 300 KLOC	Крупный проект, большая команда разработчиков, значительный объем инноваций, значительная нестабильность внешней среды.
Сблокированный	50–300 KLOC	Средний проект с инновациями, небольшая команда. Незначительная нестабильность внешней среды.
Органичный	До 50 KLOC	Некрупный проект, небольшая команда, для которой нехарактерны нововведения. Стабильная внешняя среда .

На базовом уровне COSOMO легко получить быструю оценку стоимости разработки, но данный уровень не учитывает различия в аппаратных ограничениях, качестве и опыте команды проекта (числа разработчиков -К), техники и средства разработки и т.д. В соответствии с (F.20), WH зависят нелинейно от размера рассчитываемого проекта и изменяются скачком при изменении режима. Рост WH при переходе на более высокий режим не всегда означает увеличение длительности (Т) выполнения проекта, вычисляемого по формуле (F.21):

$$WH = a \times (KLOC)^b \text{ (F.20),}$$

$$T = c \times WH^d \text{ (F.21),}$$

$$K = \frac{WH}{T} \text{ (F.22)}$$

Таблица 18. Значения коэффициентов COSOMO

Режим	Коэф. a	Коэф. b	Коэф. C	Коэф. d
Внедренный	3,6	1,20	2,5	0,32
Сблокированный	3.0	1,12	2,5	0,35
Органичный	2,4	1,05	2,5	0,38

На более высоких уровнях модель COSOMO усложняется, обрастает дополнительными коэффициентами, позволяющими повысить точность оценок. Модель позволяет проводить калибровку на основе исторических данных по осуществленным проектам.

Из-за устаревания модели COSOMO в 1997 г была представлена модель COSOMO II. COSOMO II наследует подход и принципы COSOMO, но адаптирована к современным методологиям разработки ПО (заложена возможность использования спиральной и итеративной модель ЖЦ в дополнении с каскадной, использовался Байесовский анализ в отличие от многофакторного регрессионного, возможно измерять размер проекта в LOC, и FP). В COSOMO II учитывается уровень зрелости предприятия и процесса разработки ПО согласно SEI CMM/CMMI[85].

COSOMO II имеет несколько различных вариантов использования, фактически это различные модели, объединенные под одним общим названием (табл. 19).

Таблица 19. Модель СОСОМО II

Модель	Характеристики
Ранней разработки	для приближенных оценок, до определения архитектуры, использует в качестве метрик LOC или FP.
Постархитектурная	детализированная модель, используется после разработки архитектуры, позволяет получить точные оценки, использует в качестве метрик LOC или FP.
Композиционная	ориентирована на проекты, создаваемые с применением современных средств разработки, использующих в качестве метрики 3-х мерные и объектные FP.

Достоинства и недостатки

- 1) широко распространённая и известная модель;
- 2) существует большое количество калькуляторов для модели.
- 3) В своей основе модель опирается на KLOC, тем самым вбирая в себя недостатки линейного подхода, хотя при помощи калибровок и поправочных коэффициентов влияние недостатков возможно снизить;
- 4) эмпирические коэффициенты СОСОМО I,II были подтверждены экспертами, однако было установлено, что в большинстве случаев невозможно получить требуемые оценки с необходимым уровнем достоверности.

1.7.14 Методика Госкомтруда

В 80-х годах в СССР Госкомтруда на основе метода СОСОМО реализовала собственные модели оценки трудоемкости и разработки ПО. В данных моделях задача оценки размера, трудоемкости и разработки программной системы были решены схожим образом.

1.9.15 Конус неопределенности

Конус неопределенности (Cone of Uncertainty – CofU) предложен Boehm В.W., в 1981 г. В области управления проектами CofU описывает изменение количества неопределенности в течение проекта. В начале проекта у проектной команды мало знаний о предстоящей работе и продукте и как следствие предварительная оценка содержит большой процент неопределенности. Со временем, когда проведены исследования и начата разработка, информации о проекте становится больше и неопределенность имеет тенденцию снижаться, а к концу проекта достигать нулевого значения.

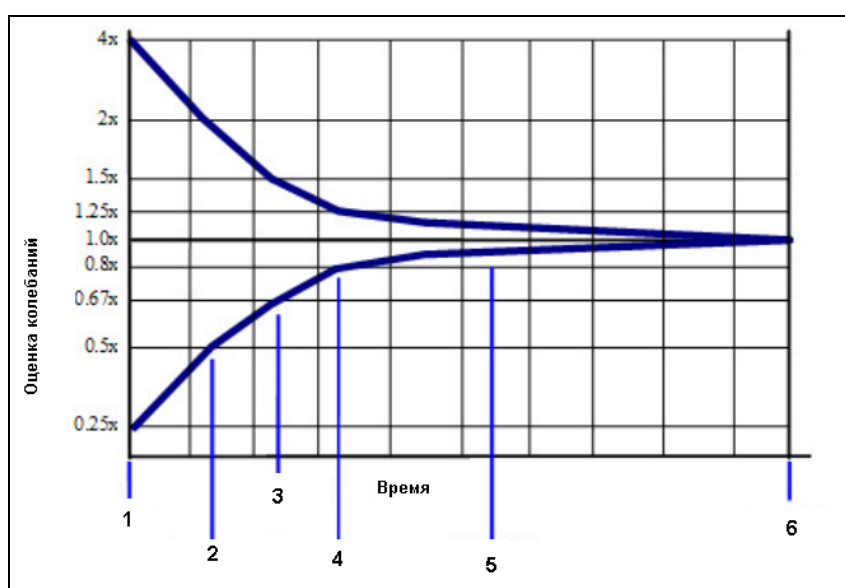


Рис.10 Конус неопределенности

Горизонтальная ось содержит основные вехи проекта, вертикальная ось содержит степень ошибок, которые могут быть совершены экспертом на различных этапах проекта. Данные оценки могут относиться ко всему содержимому проекта - размер усилий, цена, свойства или их комбинация. Оценки, сделанные на стадии инициации, могут быть не точными порядка четырех раз, как в большую, так и в меньшую сторону. Примеры реальных данных приведены на следующих рисунках (см. рис. 11,12,13,14).

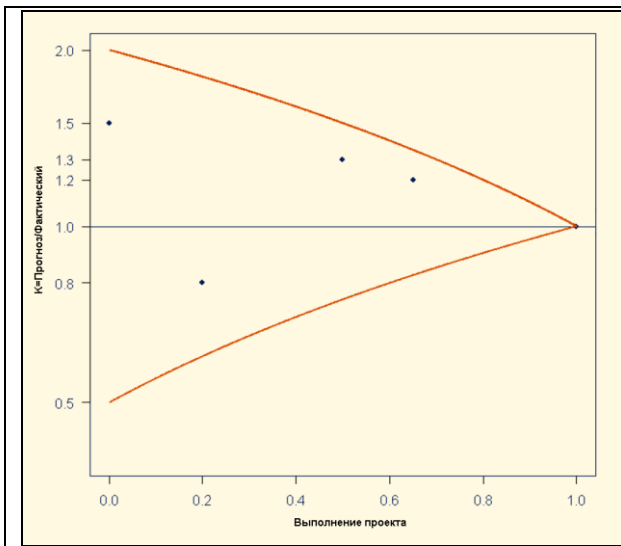


Рис.11 Конус неопределенности №1

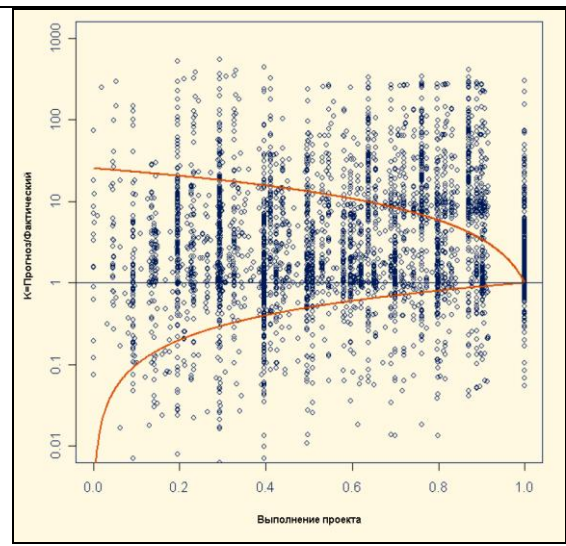


Рис.12 Конус неопределенности №2

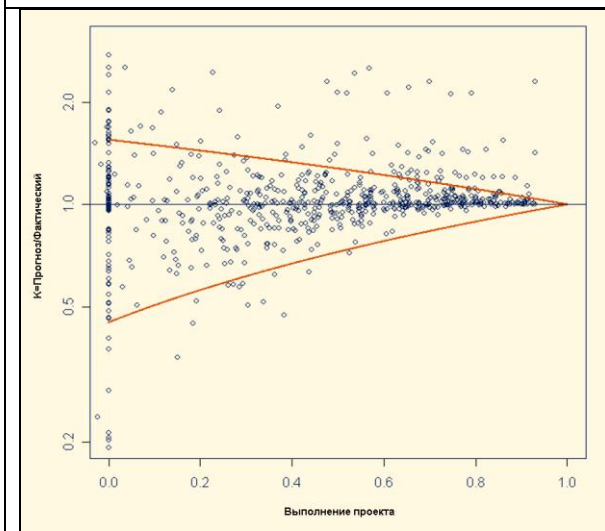


Рис.13 Конус неопределенности №3

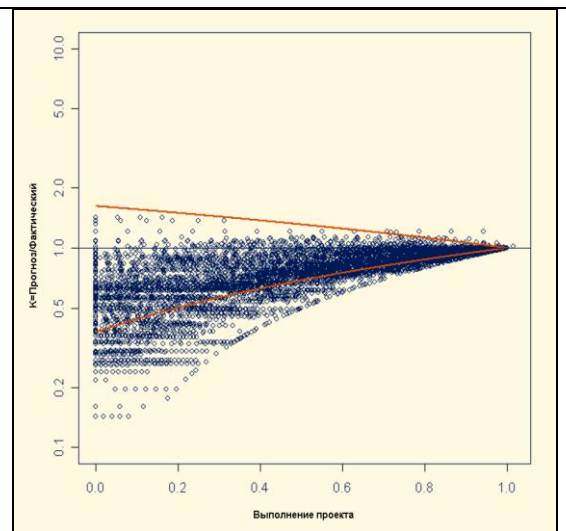


Рис.14 Конус неопределенности №4

1.9.16 Метод ДеМарко

Эффективный и простой метод оценки параметром программного проекта, на основе накопленного опыта, был предложен DeMarco T. в 1982 г. Основное использование “бэнг-метрики”, близкой по своей сути к FP. Существенная особенность метода состоит в корректировке оценок по историческим данным по ранее выполненным проектам.

1.9.17 Фактор оценки качества первоначального прогноза

Фактор оценки качества первоначального прогноза (Estimating Quality Factor - EQF), предложен DeMarco Т. в книге “Controlling software project” в 1982 г.. EQF является мерой отклонения прогнозируемого значения от реального, позволяет измерять качество предварительной оценки. Рассмотрим на примере использование EQF. Предположим, что проект стартовал 01.01.2011 и закончится 01.09.2011. Суммарная стоимость проекта составит 20 миллионов. Во время выполнения проекта были проведены 4 прогноза (табл. 20)

Таблица 20. Предварительный прогноз

Прогнозируемая стоимость	Дата прогноза	Продвижение проекта (%)	Коэффициент $K=F/A$ (Forecast/Actual)
30	01.01.2011	0	1.5 =(20/30)
16	19.02.2011	20	0.8
26	02.05.2011	50	1.3
24	08.06.2011	65	1.2

Для этой таблице построим график зависимости $K = F/A$ (Forecast/Actual,) от процента продвижения проекта. Для каждой оценки посчитаем отклонения от реального значения, вычисляя площадь прямоугольника между оценкой и реальным значением (рис. 15). Рассчитаем EQF как обратную величину к суммарной площади. $EQF=1/(0.1+0.006+0.045+0.07)=3.6$.

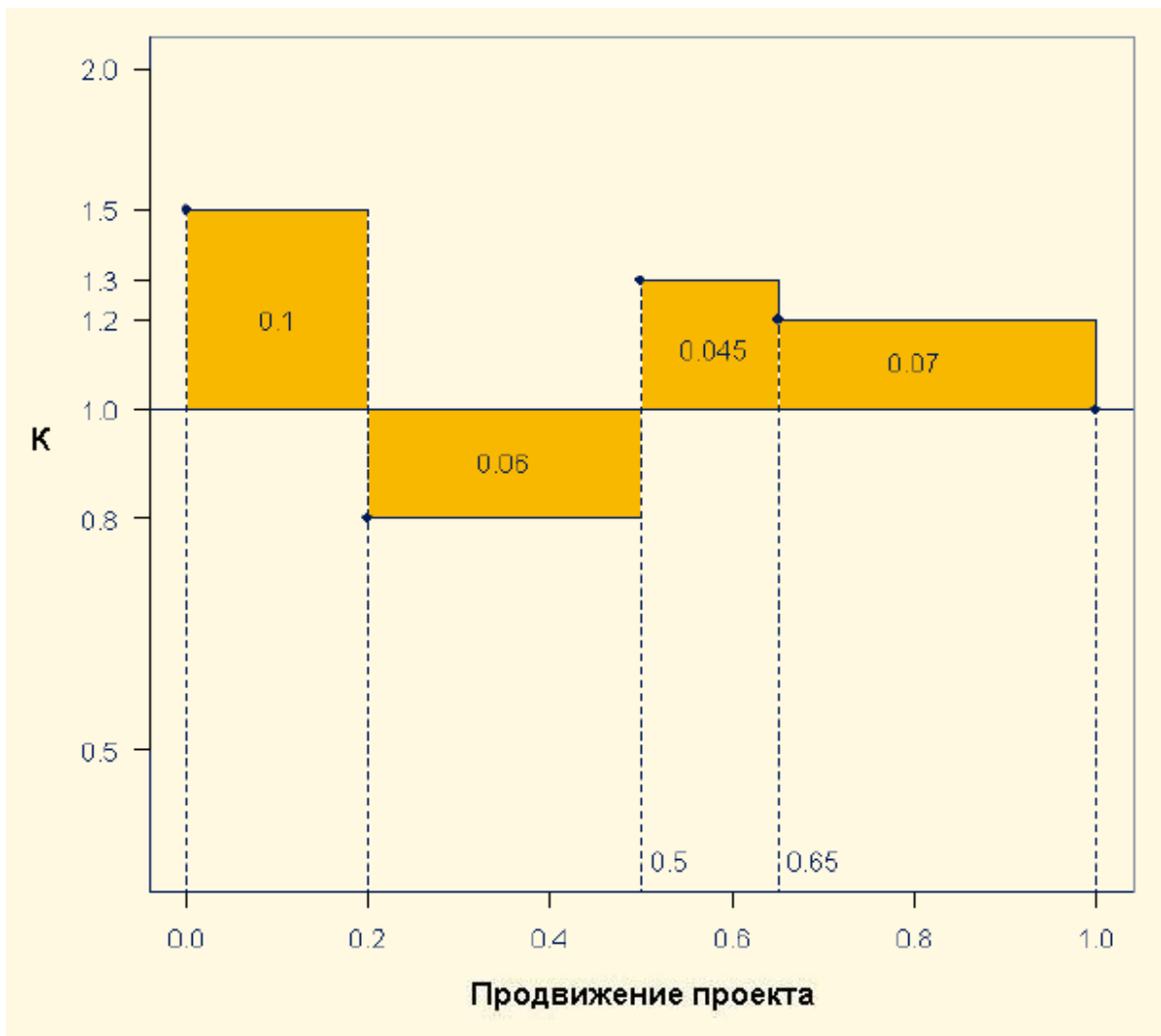


Рис.15 отклонения от реального значения

Исходя из определения EQF:

- ✓ для любого проекта становится возможным его расчет;
- ✓ низкое значение EQF соответствует тому, что все прогнозы были низкого качества и отклонение первоначальных прогнозов велико;
- ✓ высокое значение EQF соответствует тому, что все прогнозы были высокого качества и отклонение первоначальных прогнозов минимально;
- ✓ с использованием EQF становится возможным провести качественную оценку прогнозирования проектов и сравнить полученные результаты.

Конус неопределенности и EQF достоинства и недостатки

- ✓ простота и прозрачность;
- ✓ позволяет понять тенденцию оценок каждого эксперта;
- ✓ обманчивые и односторонние определения;
- ✓ искажение практической оценки;
- ✓ бессмысленные данные;
- ✓ коническая форма конуса не является основанием улучшения оценки, но может быть использована как семейство распределений с ожидаемой точностью и прогнозируемым уклоном для дальнейшего прогноза.

Построение конуса неопределенности, а по сути отношения прогнозируемой величины к актуальной (Forecast/Actual) показывает потенциальную предвзятость эксперта вовлеченного в прогноз. Примеры реальных данных приведены на следующих рисунках (см. рис. 10,11,12,13). Распределение отношения прогноза к фактическому значению изменяются между организациями, по крайней мере, в трех измерениях: в точности оценки, в тенденции прогнозов сходиться, и в систематической предвзятости эксперта.

Выводы по первой главе

Исходя из анализа проведенного выше можно говорить о том, что эмпирические расчеты и коэффициенты различных моделей нашли свое подтверждение, но в тоже время остается проблема – в большинстве случаев не удается получить оценки длительности разработки ПО с приемлемым уровнем достоверности. Этот факт говорит о существующих серьезных проблемах в области управления ИТ - проектами и оценки длительности разработки ПО. ИТ индустрия – это сложная предметная область, в которой результат зависит о ряда объективных и субъективных факторов, которые трудно поддаются формализации и оценке.

Brooks F.P. в своей статье The mythical Man-Month пишет: “создание программных систем всегда будет трудным”[20], исход из этого можно утверждать, что оценка этого процесса будет не менее трудной [26]. “Оценка трудоемкости ИТ- проектов должна быть вероятностным утверждением, для нее существует некоторое распределение вероятности, которое может быть широким, если неопределенность высокая или узким, если неопределенность низкая. Использование собственного опыта и опыта коллег, полученного в схожих проектах, это наиболее прагматичный подход, позволяющий получить реалистичные оценки трудоемкости и срока реализации программного проекта. Если собственный опыт аналогичных проектов отсутствует, а коллеги-эксперты недоступны, необходимо использовать формальные методики, основанные на обобщенном отраслевом опыте. Нереалистичность полученных оценок - серьезный демотивирующий фактор для участников проектной команды. Недооценка приводит к ошибкам планирования и неэффективному взаимодействию, авральные сроки, психологическое давление, сверхурочные, служат причиной того, что затраты на проект растут неограниченно и экспоненциально”[13,88].

Глава 2. Создания модели оценки

2.1. Формализация модели

По проведённому в первой главе анализу сформированы требования к методу:

- ✓ применимость на практике;
- ✓ время и точность оценки должны удовлетворять требованиям оперативного управления;
- ✓ независимость от объёма программных проектов и выбранных методологий управления ими;
- ✓ гибкий учёт экспертной оценки;
- ✓ “прозрачность” для руководства;
- ✓ учёт достоинств и недостатков существующих моделей и методов.

2.2. Модель оценивания

Исходя из анализа, проведённого в первой главе, за основу метода выберем следующие подходы и средства реализации:

- ✓ методику PERT для первого шага алгоритма;
- ✓ принцип Wideband Delphi для построения методики оценки;
- ✓ CofU и EQF фактор для анализа распределения оценок эксперта.
- ✓ MS Project в качестве базового программного обеспечения;
- ✓ алгоритм ДОПС;
- ✓ программную надстройку над MS Project, для реализации алгоритма оценки.

2.3. Анализ оценок экспертов

В главе собраны и проанализированы данные экспертных оценок по шести программным проектам (или их отдельным законченным фазам) в которых автор выступал в различных ролях от эксперта и технолога до руководителя проекта (табл. 22).

Таблица 22. Характеристики проанализированных проектов

№	Продолжительность, (раб.дней)	Количество оценок за весь проект	Количество экспертов всего
1	256	98	4
2	198	102	3
3	427	101	8
4	987	603	47
5	571	389	29
6	158	248	12

На каждом этапе проекта экспертам предлагалось оценить итоговый срок разработки проекта в днях[41]. На первых трёх проектах проиллюстрируем анализ полученных данных, для последних трёх приведём итоговый результат.

Для первых трех проектов построим конус неопределённости для фиксированного эксперта. Для этого по таблице замеров построим график зависимости $K = F/A$ (отношение прогнозируемой величины к актуальной – $K = F/A$; (Forecast/Actual),) от процента продвижения проекта (. рис. 16-24).

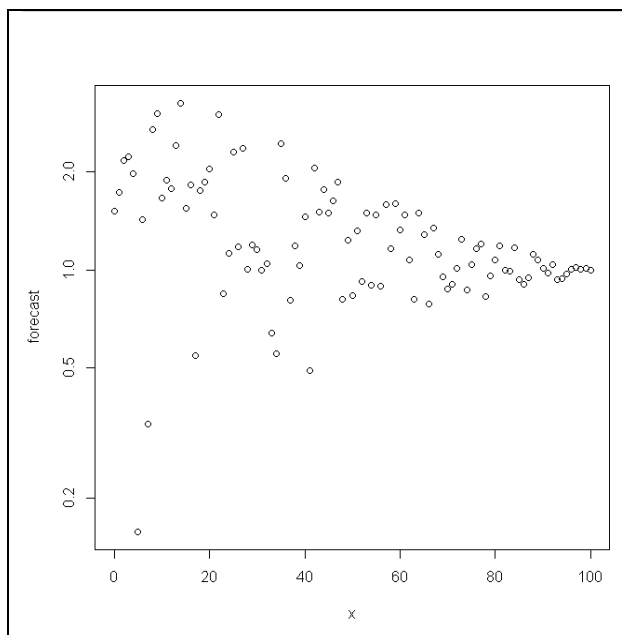


Рис 16. Первый проект,

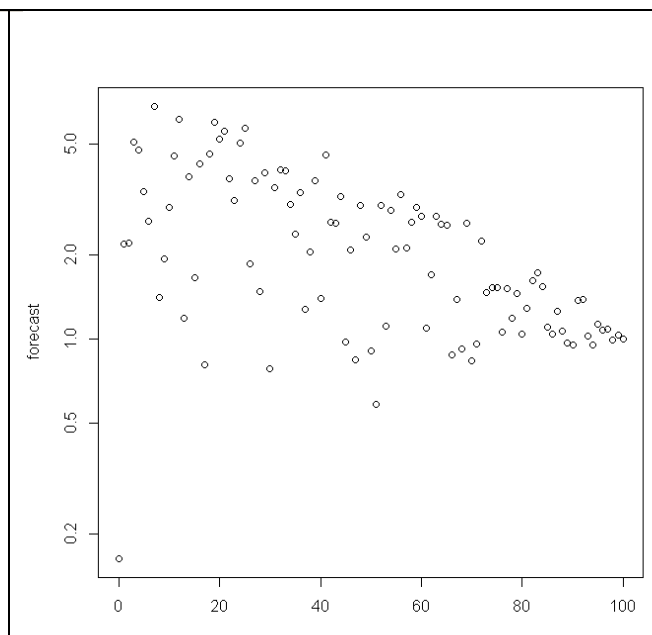
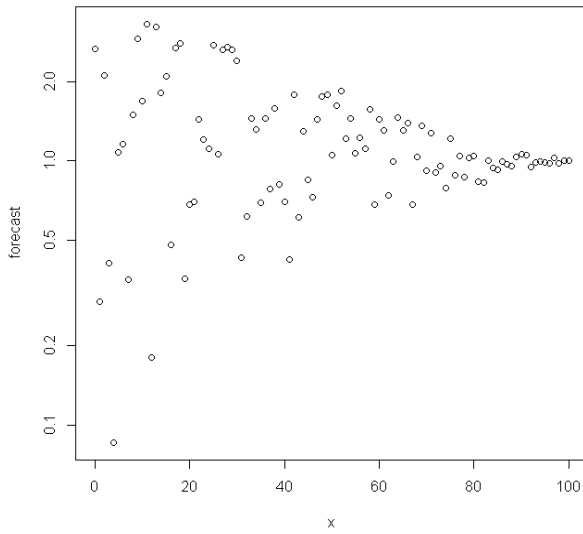


Рис 17. Первый проект,

K_T эксперта А



K_T эксперта В

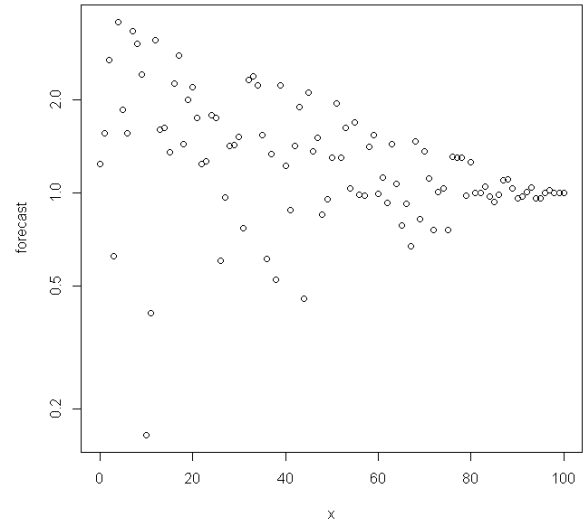


Рис 18. Первый проект,

K_T эксперта С

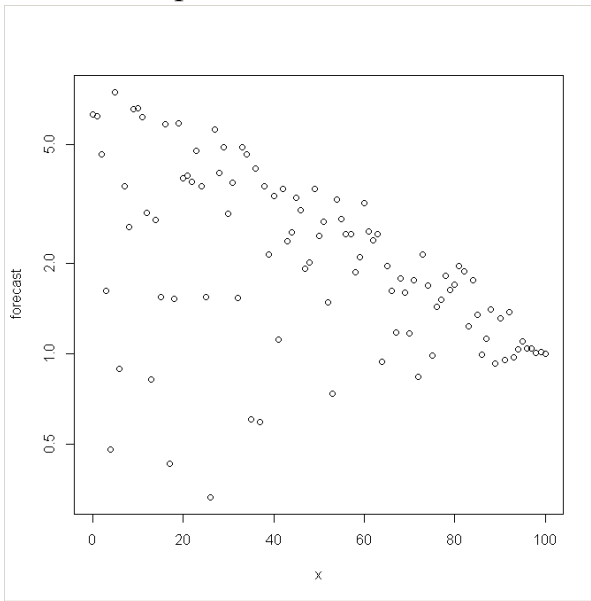


Рис 19. Второй проект,

K_T эксперта А

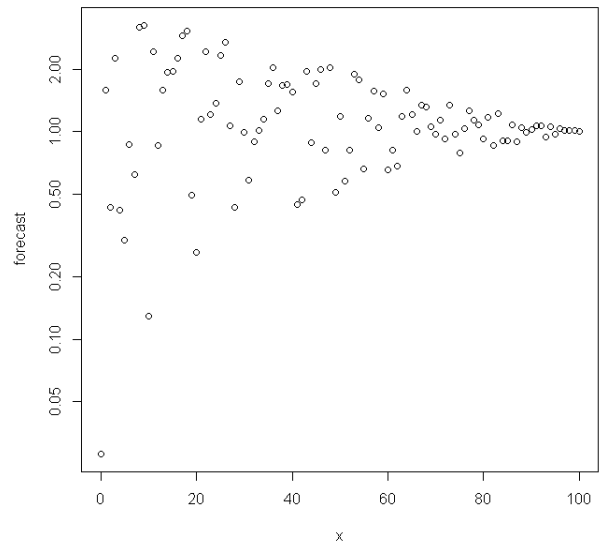


Рис 20. Второй проект,

K_T эксперта В

Рис 21. Второй проект,

K_T эксперта С

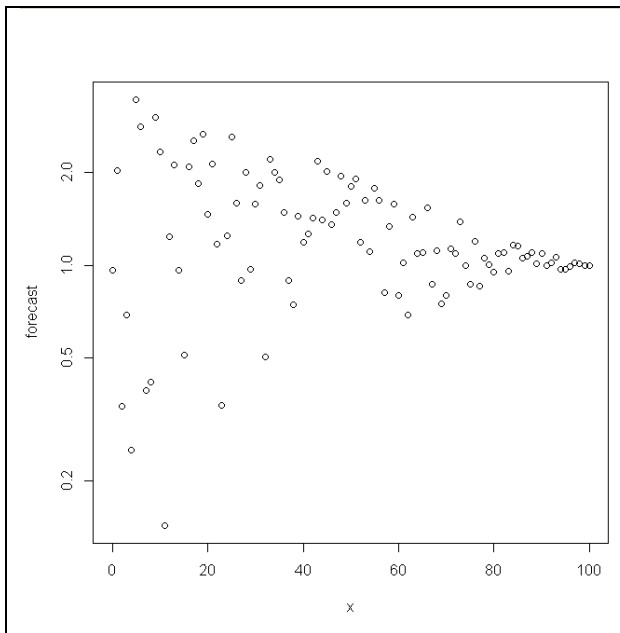


Рис 22. Третий проект,
 K_T эксперта А

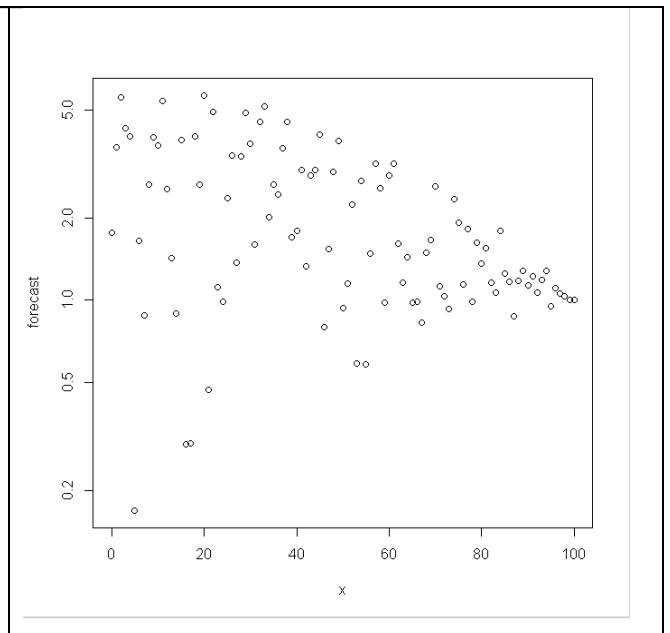


Рис 23. Третий проект,
 K_T эксперта В

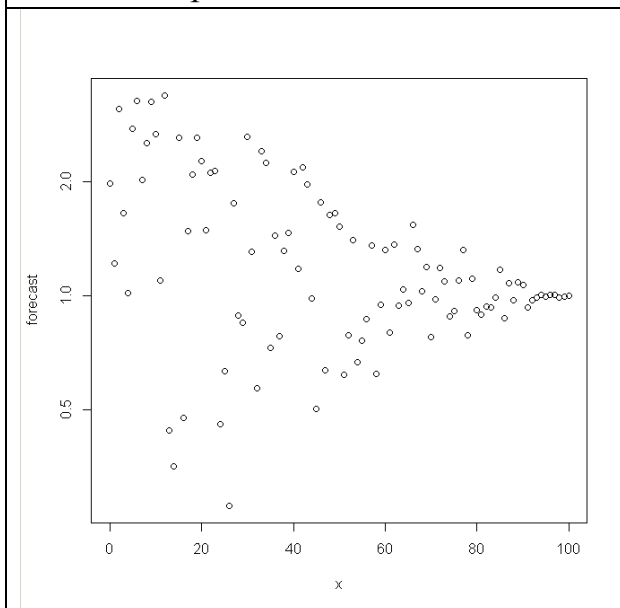


Рис 24. Третий проект,
 K_T эксперта С

По результатам первых трех проектов:

- 1) Построим функции распределения оценок экспертов по проектам.
- 2) Используя метод максимального правдоподобия [34], найдем оценки математического ожидания и стандартного отклонения μ и σ .

3) На едином графике построим эмпирическую функцию распределения, логнормальное распределение и бета - распределение оценок экспертов по проектам.

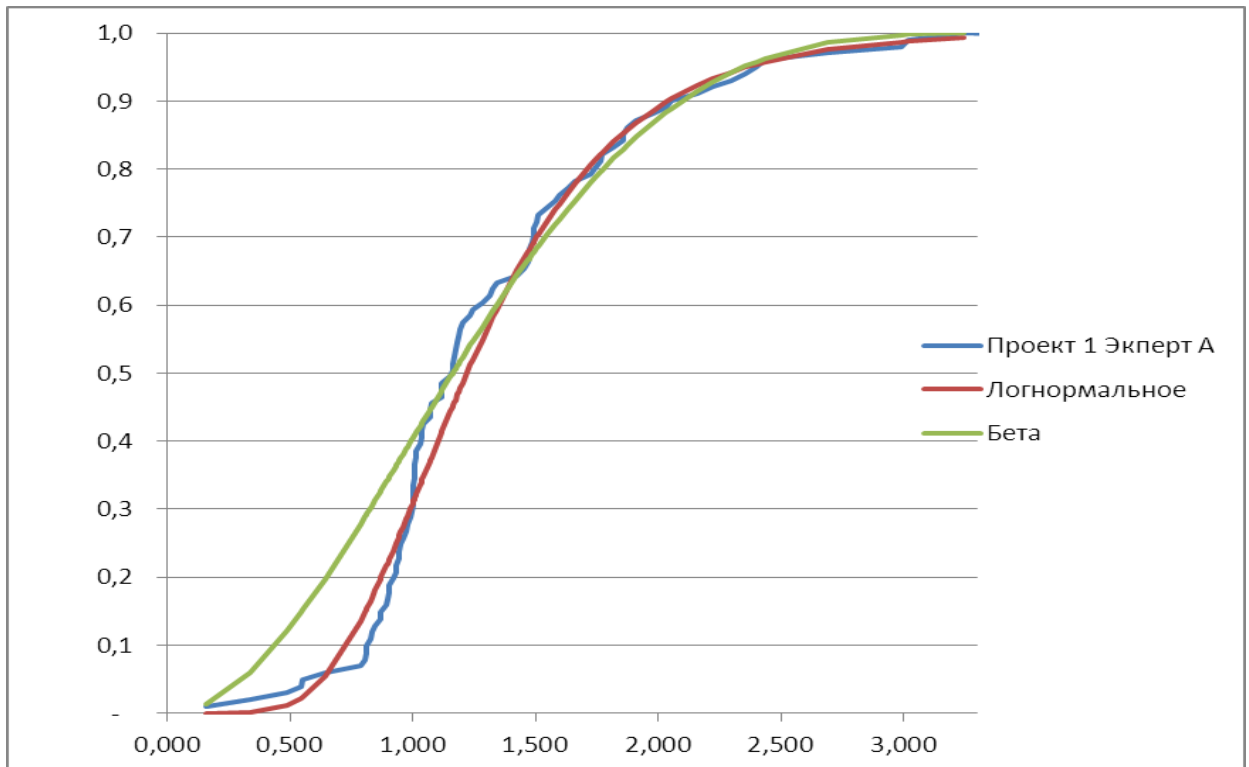


Рис 25. Функция распределения Проект 1, Эксперт А

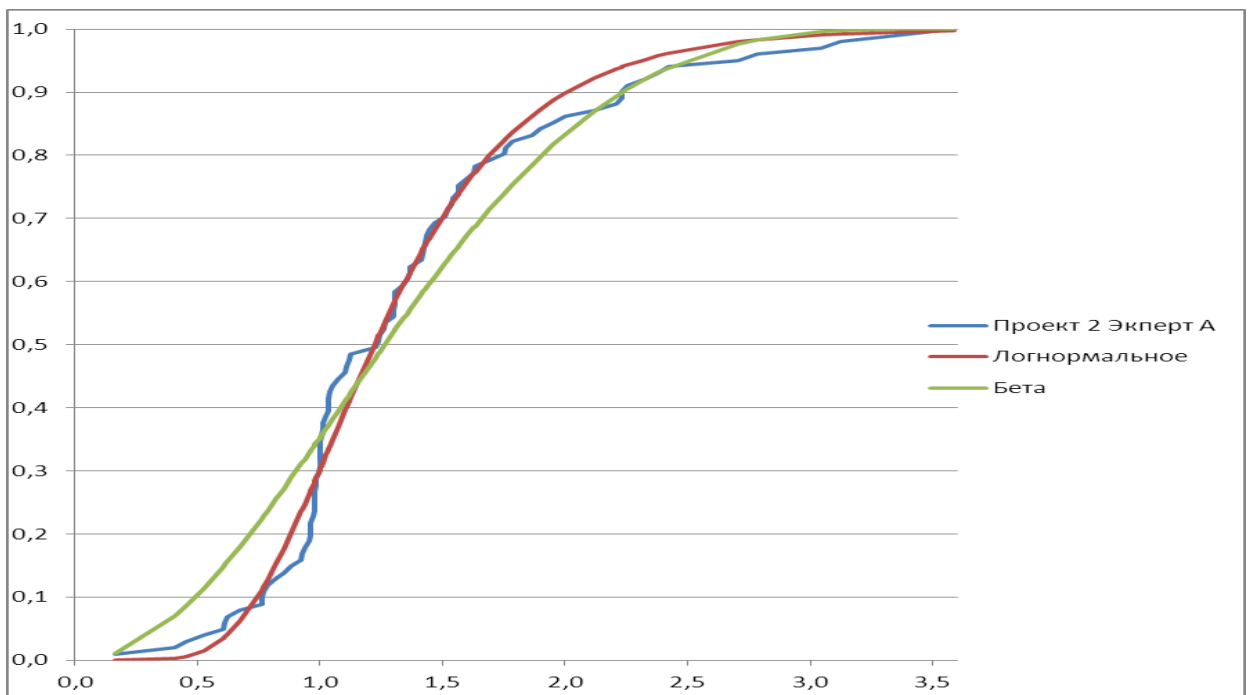


Рис 26. Функция распределения Проект 2, Эксперт А

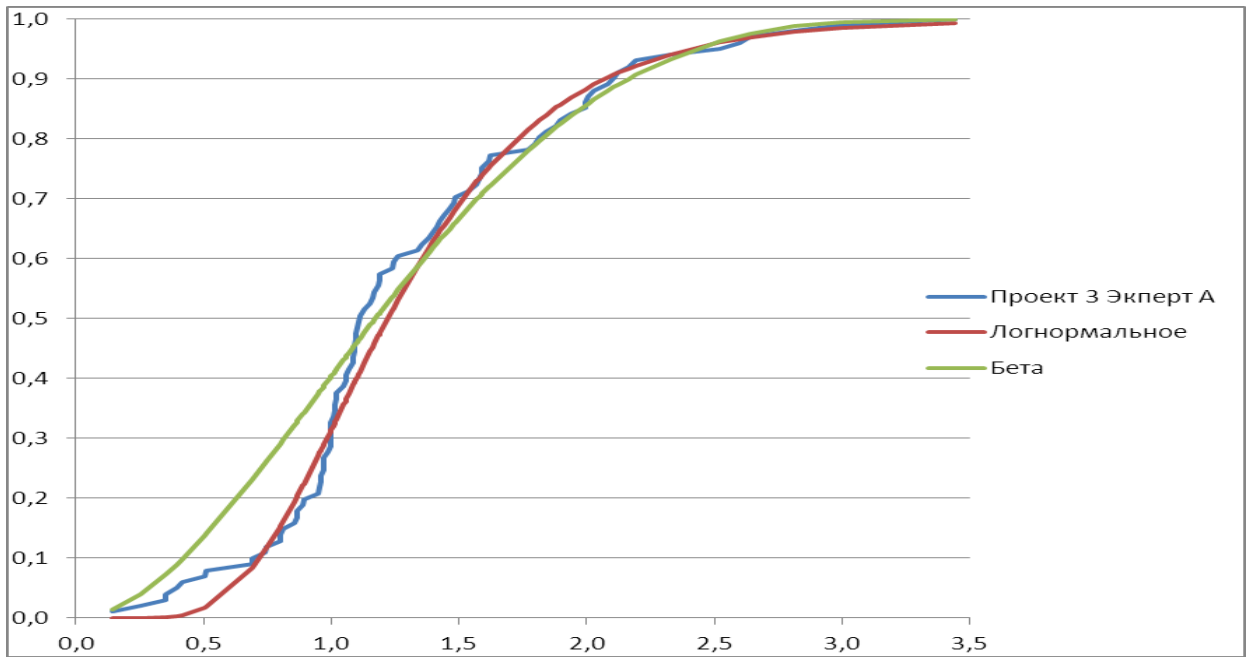


Рис 27. Функция распределения Проект 3, эксперт А

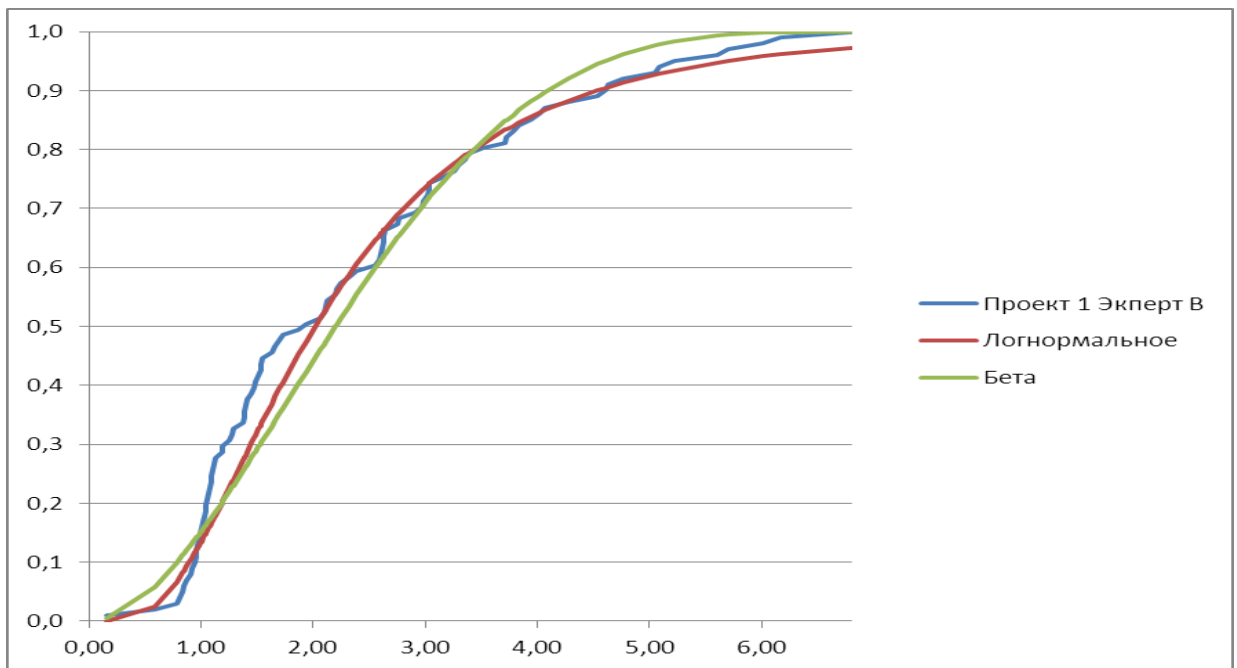


Рис 28. Функция распределения Проект 1, Эксперт В

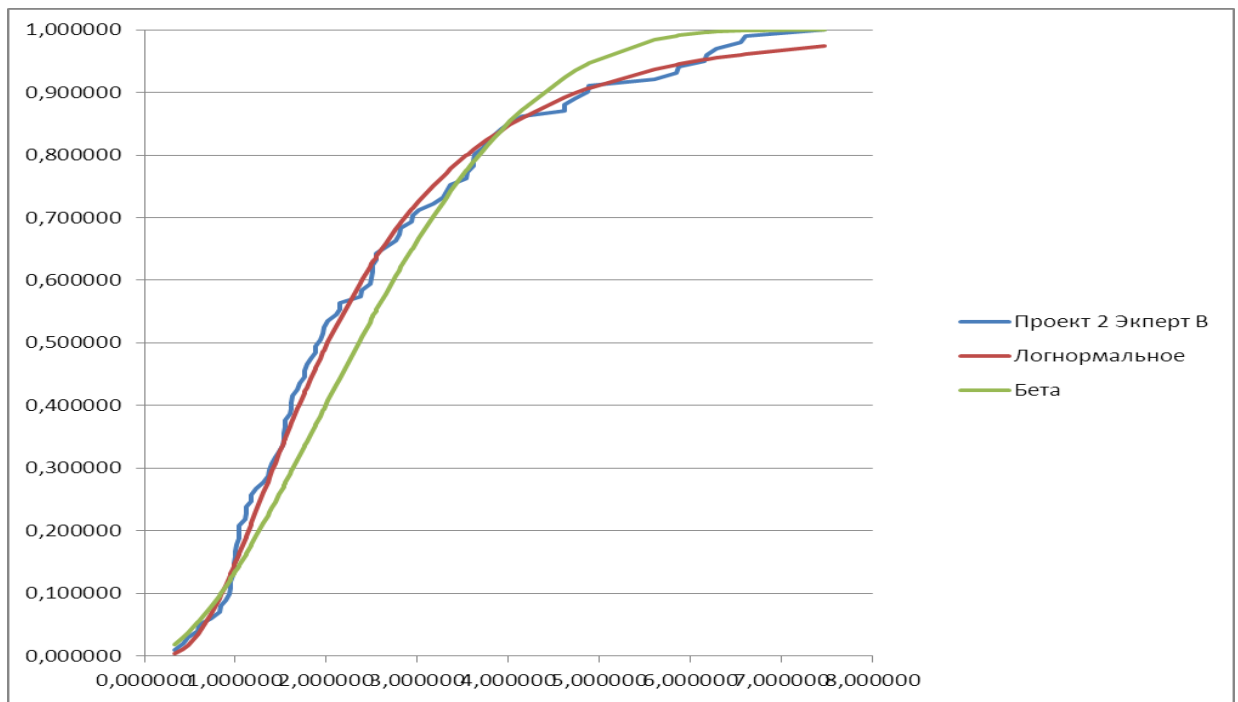


Рис 29. Функция распределения Проект 2, Эксперт В

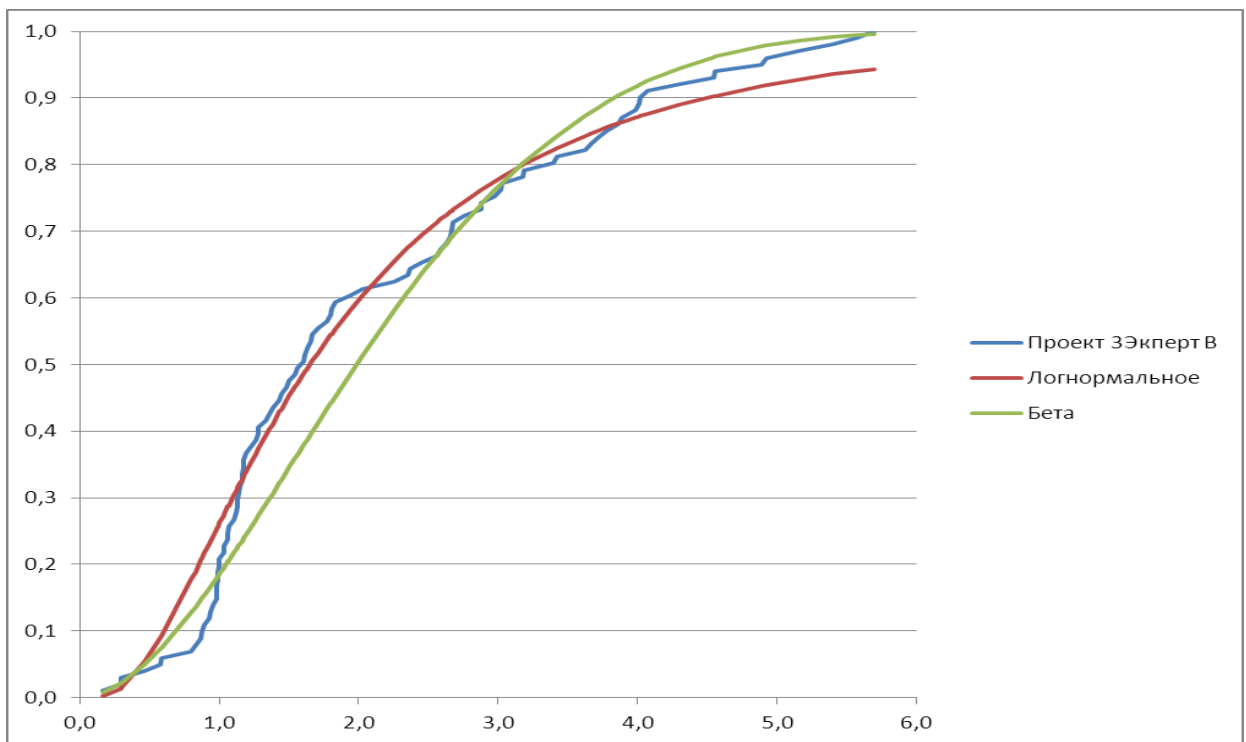


Рис 30. Функция распределения Проект 3, Эксперт В

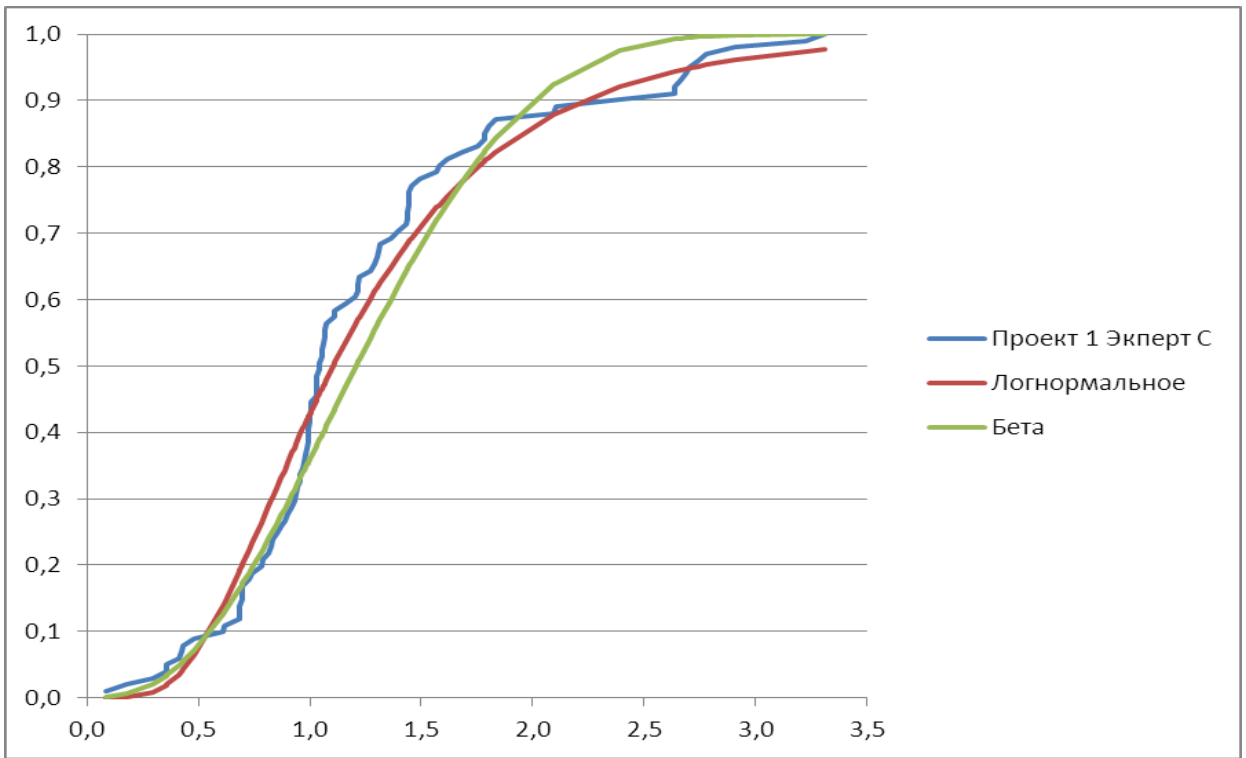


Рис 31. Функция распределения Проект 1, Эксперт С

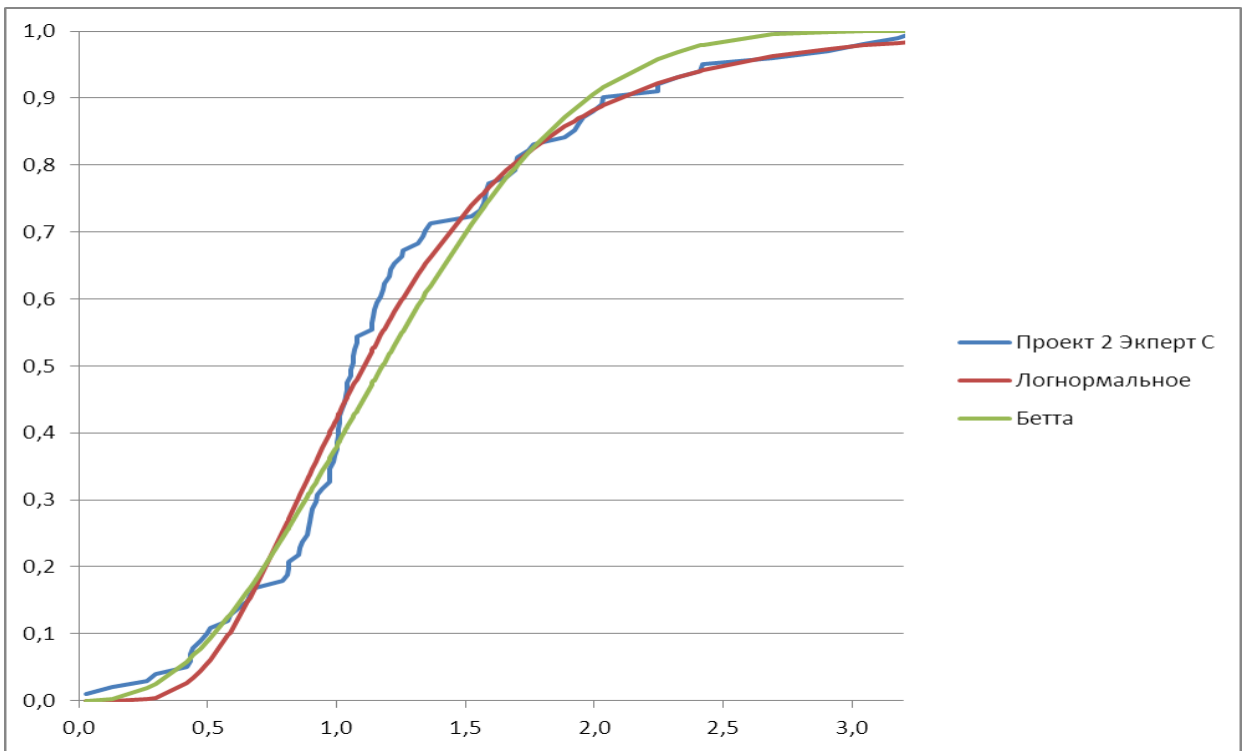


Рис 32. Функция распределения Проект 2, Эксперт С

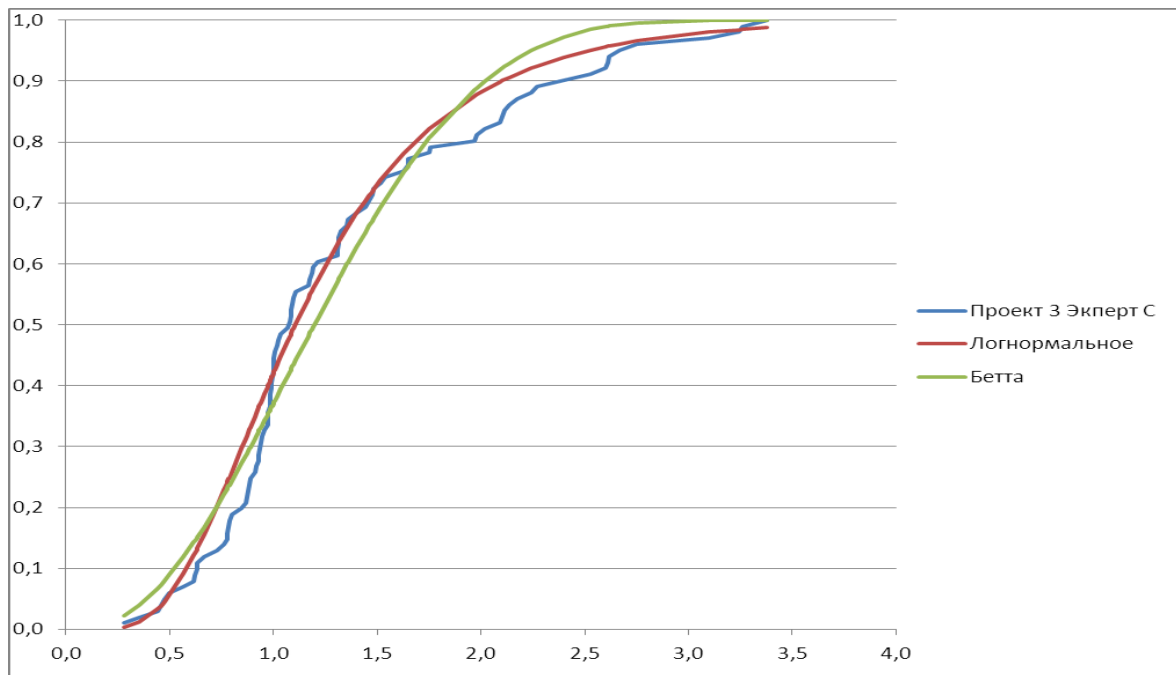


Рис 33. Функция распределения Проект 3, Эксперт С

Таблица № 23. Характеристики логнормального распределения эксперта А по проектам

Эксперт А	Первый проект	Второй проект	Третий проект
Мат. Ожидание- μ	0.2	0.2	0.2
Стандартное отклонение - σ	0.4	0.39	0.4

Таблица № 24. Характеристики бета - распределения эксперт А по проектам

Эксперт А	Первый проект	Второй проект	Третий проект
α	2,15	2,2	2
β	4	3.8	3.8
Мат. Ожидание- μ	0,35	0,37	0,34
Стандартное отклонение - σ	0,18	0,18	0,18

Таблица № 25. Характеристики Логнормального распределения Эксперт В по проектам

Эксперт В	Первый проект	Второй проект	Третий проект
Мат. Ожидание- μ	0.7	0.7	0.5
Стандартное отклонение - σ	0.63	0.67	0.78

Таблица № 26. Характеристики Бета-распределения Эксперт В по проектам

Эксперт В	Первый проект	Второй проект	Третий проект
α	2	2	2
β	4	4	2.5
Мат. Ожидание- μ	0,34	0,34	0,29
Стандартное отклонение - σ	0,18	0,18	0,16

Таблица № 27. Характеристики Логнормального распределения Эксперт С по проектам

Эксперт С	Первый проект	Второй проект	Третий проект
Мат. Ожидание- μ	0.1	0.1	0.1
Стандартное отклонение - σ	0.55	0.5	0.5

Таблица № 28. Характеристики Бета-распределения Эксперт С по проектам

Эксперт С	Первый проект	Второй проект	Третий проект
α	2.9	2.8	2.7
β	5	5	4.7
Мат. Ожидание- μ	0,37	0,36	0,36
Стандартное отклонение - σ	0,16	0,16	0,17

2.4. Анализ распределения оценок экспертов

Критерия согласия Пирсона [34]

Пусть K_T имеет эмпирическое распределение $F^*(K_T)$.

$$K_T^n = (x_1, \dots, x_n), x_i \in [a, b], \forall i = 1..n$$

Выдвинем гипотезу H_1 - Случайная величина K_T имеет бета распределение.

Выдвинем гипотезу H_2 - Случайная величина K_T имеет логнормальное распределение.

Проведём сравнение эмпирического $F^*(K_T)$ и теоретического распределения $F_1(K_T)$ с помощью критерия согласия Пирсона (критерий χ^2).

Посчитаем значение статистики χ^2 . Число степеней свободы выберем равное 7. Зададим два уровня значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$. Критическое значение критерия согласия Пирсона для уровня значимости α_1 и заданного числа степеней свободы равно $D_{0,10} = 12,01704$, α_2 и заданного числа степеней $D_{0,05} = 14,06714$ [35].

Таблица № 29. Значение статистики χ^2 критерия согласия Пирсона

Эксперт А	Гипотеза H_2	Гипотеза H_1
Проект 1	31,3547	69,46395
Проект 2	29,7356	71,80417
Проект 3	18,1001	46,44541
Проект 4	28,7256	70,80317
Проект 5	18,5601	46,89417
Проект 6	16,5684	68,43177
Эксперт В		
Проект 1	25,06886	37,71238
Проект 2	4,383424	14,80502
Проект 3	29,09336	40,29234
Проект 4	9,373724	18,80808
Проект 5	28,58001	39,29784
Проект 6	2,589001	17,53545
Эксперт С		
Проект 1	34,84151	41,27366
Проект 2	10,2019	14,68908
Проект 3	31,37954	47,49719
Проект 4	11,34724	15,56321
Проект 5	32,83131	39,25764
Проект 6	9,577993	19,45789

Значение критерия согласия Пирсона для гипотезы H_2 всегда меньше значения критерия согласия Пирсона для гипотезы H_1 .

В шести случаях из пятнадцати гипотеза H_2 принимается для уровня значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$, в пятнадцати случаях из пятнадцати гипотеза H_1 отвергается для обоих уровней значимости.

Проведем дополнительное исследование, используем **критерий согласия типа Колмогорова-Смирнова**[36].

Посчитаем значение статистики D_n . Выберем два уровня значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$. Критическое значение критерия согласия Колмогорова-Смирнова для уровня значимости α_1 и заданного числа степеней свободы равно $D_{0,10} = 0,12067$, α_2 и заданного числа степеней $D_{0,05} = 0,13403$

Таблица № 30. Значение статистики D_n критерия согласия Колмогорова-Смирнова.

Эксперт А	Гипотеза H_2	Гипотеза H_1
Проект 1	0,089830	0,209607
Проект 2	0,086660	0,152819
Проект 3	0,099081	0,166192
Проект 4	0,076590	0,172714
Проект 5	0,069686	0,161122
Проект 6	0,099088	0,168192
Эксперт В		
Проект 1	0,106070	0,138456
Проект 2	0,046558	0,140876
Проект 3	0,118058	0,145065
Проект 4	0,103590	0,142414
Проект 5	0,006786	0,165152
Проект 6	0,001805	0,130066
Эксперт С		
Проект 1	0,084056	0,153932
Проект 2	0,085433	0,152896
Проект 3	0,104994	0,142139
Проект 4	0,074753	0,153735
Проект 5	0,085433	0,172576
Проект 6	0,054753	0,163778

Значение критерия согласия Колмогорова-Смирнова для гипотезы H_2 всегда меньше значения критерия согласия Колмогорова-Смирнова для гипотезы H_1 .

При уровне значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$. гипотеза H_2 принимается во всех случаях.

При уровне значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$. гипотеза H_1 отвергается для обоих уровней значимости.

Программный проект будем называть *крупным* если оценочное время его выполнения более одного года при количестве исполнителей более 20 человек.

Крупный программный проект будем называть *определенным* в рамках метода динамической оценки, если он выполняется в организации с высоким уровнем зрелости (3 и выше по шкале СММ) с использованием итеративных моделей разработки.

Утверждение: Эмпирическое распределение $F^*(K_T)$ выполненных экспертом оценок времени фаз определенного программного проекта ближе к логнормальному распределению, чем к бета распределению.



1.Анализируемые эмпирические данные были выбраны произвольным образом среди *определенных* программных проектов и случайных экспертов.

2. Значение критерия согласия Пирсона для гипотезы H_2 всегда меньше значения критерия согласия Пирсона для гипотезы H_1 .

3. Анализ исходных данных при помощи критерия согласия типа **Колмогорова-Смирнова** при уровне значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$ показал, что значение критерия согласия **Колмогорова-Смирнова** для гипотезы H_2 всегда меньше значения критерия согласия Пирсона для гипотезы H_1 .

4.При уровне значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$. гипотеза H_2 принимается во всех случаях.

5. При уровне значимости $\alpha_1 = 0,10, \alpha_2 = 0,05$. гипотеза H_1 отвергается для обоих уровней значимости.



Учтём этот факт при построении модели оценивания, изменим алгоритм расчёта формулы (F.15).

2.5. Построение математической модели

Управление проектом можно представить в виде автоматизированной системы, в которой рутинные процессы сбора и анализа информации выполняются автоматически, но управляет всей системой руководитель проекта, который принимает решения. Структурная схема системы управления представлена следующим образом:

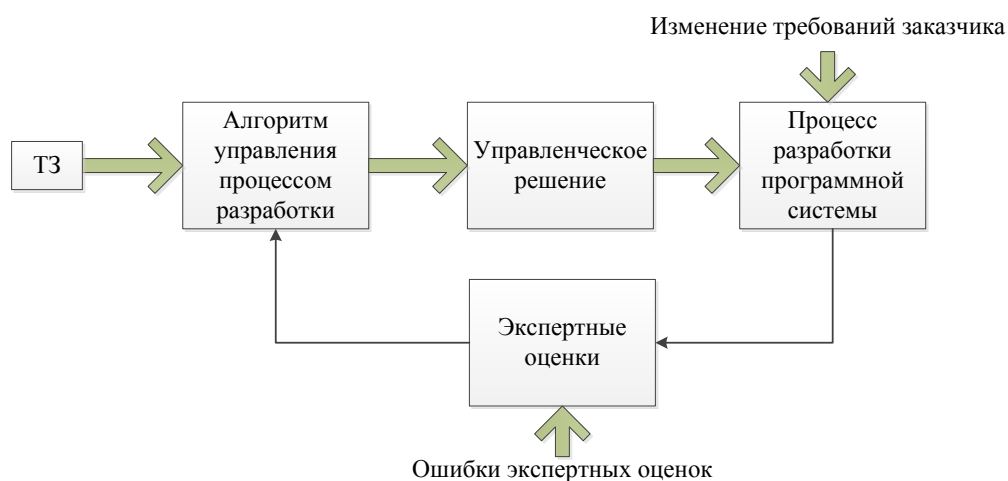


Рис. 34 Структурная схема системы управления

Введем основные понятия и определения

Эксперт – специалист, член рабочей группы проекта или команды заказчика, привлекаемый для оценок состояния процесса разработки в целях определения текущего состояния, сроков, качества и стоимости работ внутри проекта.

Экспертное оценивание – процесс получения оценки задачи на основе различных мнений экспертов для последующего принятия управленческого решения.

Пакет работ – элемент работ проекта, находящийся на самом низком уровне каждого ответвления ИСР. Обозначим пакет работ через P_i , длительность пакета работ – TP_i где i – номер пакета.

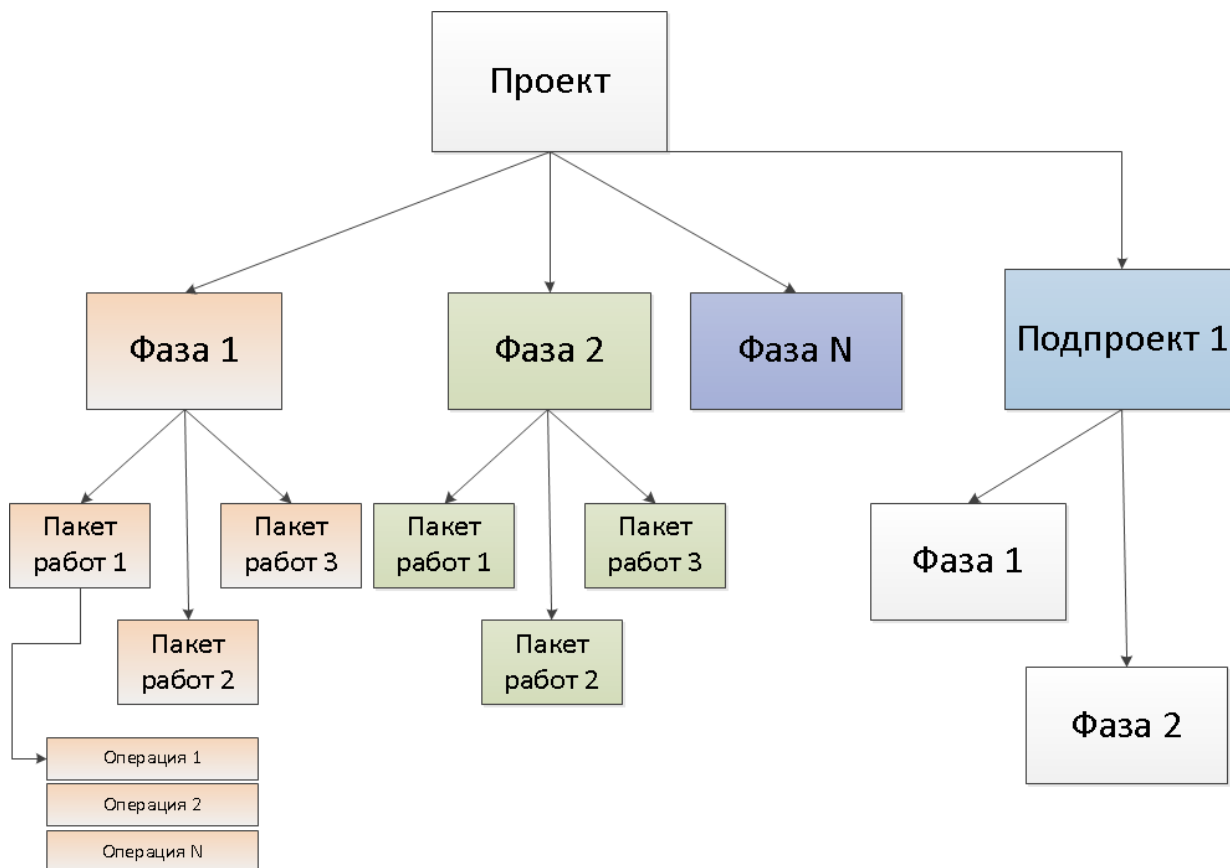


Рис. 35 Структура проекта.

Фаза проекта - Ряд логически связанных операций проекта, завершающихся достижением основных результатов. Фаза проекта состоит из пакетов работ.

Обозначим фазу проекта через $F_n = \{ \dots \}$, где N – количество фаз проекта.

Активная фаза проекта – фаза проекта, внутри которой существует хотя бы один выполняемый пакет работ P_i . В противном случае фаза **неактивна**.

Обозначим активную фазу проекта через FA_m , неактивную через FP_m , где m – номер фазы. Соответственно, длительности фаз обозначим TFA_m и TFP_m .

Обозначим:

1) $\{V_1, V_2, \dots, V_k\}$ – множество экспертов проекта;

2) $\alpha_{m,P_i,k}$ – коэффициент доверия в пакете работ i фазы m , для эксперта V_k ,

причём $\sum_{k=1}^N \alpha_{m,P_i,k} = 1, i = const$;

3) $Y_{i,k} = (V_k, \alpha_{m,P_i,k})$ – уникальная пара для пакета работ i ;

4) $Pt_i = \frac{A_i + B_i + 4 * M_i}{6}$ [A.1], – время выполнения пакета работ i по методу PERT;

$\hat{t}_{m,P_i,k}$ – экспертная оценка времени выполнения пакета работ i эксперта k фазы m ;

5) $Z_m(t) = \sum_{i=1}^H TP_i + \sum_{k=1}^{G-H} \alpha_{m,P_i,k} \times \hat{t}_{m,P_i,k}$ [A.2] – длительности фазы m в момент

времени t , где H – количество выполненных работ внутри фазы m . G – общее количество работ внутри фазы m .

6) $T = T(t)$ – длительность критического пути, где t – произвольный момент времени, $t \in (t_c, +\infty)$, t_c – время старта проекта.

Утверждение. Если t_c – время старта проекта, то в момент времени t длительность критического пути может быть найдена следующим образом:

1) $t \leq t_c$

$$T(t) = \sum_{m=1}^N TFP_m(\hat{t}_m, \hat{d}_m), \text{ где}$$

\hat{t}_m – оценочное время начала фазы m по методу PERT, стоящей на критическом пути;

\hat{d}_m – оценочная длительность фазы m по методу PERT, стоящей на критическом пути, $\hat{d}_m = \sum_i^w Pt_i$, где W – количество неактивных пакетов работ фазы i ;

Pt_i – длительность пакетов работ внутри фазы m .

2) $t \geq t_c$

$T(t) = \sum_{m=1}^N TFP_m(t, \hat{d}_m) + \sum_{m=1}^N TFA_m(t, \hat{d}_m)$, где функции

$TFP_m(t, \hat{d}_m)$ и $TFA_m(t, \hat{d}_m)$ определяются следующим образом:

$$TFP_m(t, \hat{d}_m) = \begin{cases} \hat{d}_m, & t < \hat{t}_m \\ 0, & t \geq \hat{t}_m \end{cases}$$

$$TFA_m(t, \hat{d}_m) = \begin{cases} \hat{d}_m, & t = \hat{t}_m \\ Z_m(t), & t \geq \hat{t}_m \\ 0, & t < \hat{t}_m \end{cases} \quad [\text{A.3}],$$

2.6. Алгоритм оценивания

Состоит из четырех основных этапов:

1. Этап предварительного анализа.

1.1. Разработать иерархическую структуру работ (ИСР); В данном процессе происходит декомпозиция целей проекта на более мелкие и более управляемые компоненты.. Пакет работ - нижний уровень ИСР.

Декомпозицию работ необходимо производить до тех пор как только станет возможно реалистично оценить сроки и ресурсы каждого пакета[8,11];

1.2. Определить взаимосвязь операций, используя метод диаграмм предшествования;

1.3. Построить сетевую диаграмму;

1.4. Разбить проект на фазы $F_n = \{F_1, F_2, \dots, F_N\}$, каждую фазу разбить на пакеты работ P_i .

2. Определение количества предполагаемых экспертов.

Для каждого пакета работ P_i фазы m найти и определить количество предполагаемых экспертов $V_{m,i}$. Для каждой фазы проекта, подпроекта, работы, находящейся на первом уровне иерархии необходимо найти и определить количество предполагаемых экспертов. Стоит отметить, что сложность для оценки представляют новые уникальные операции. Для каждой задачи, находящейся на нижнем уровне иерархии, достаточно оценок 3-5 экспертов. По мере детализации и роста ИРС в глубину количество предполагаемых экспертов может быть скорректировано. Выбранных экспертов необходимо включить в состав проектной команды, разграничить роли и обязанности.

3. Начальный этап оценки проекта.

Найти:

3.1 Длительности всех пакетов работ P_i проекта согласно [A.1];

3.2 Критический путь;

3.3 Оценку длительности критического пути проекта согласно [A.3];

3.4 Резервное время, критическую деятельность, время ввода, время запаздывания, резерв времени.

Критический путь - наиболее длинный возможный непрерывный путь, от начала события до его полного завершения. Данный путь определяет количество календарного времени, необходимого для выполнения всего проекта. Задержки в работах на критическом пути ведут к задержкам в достижении конечного события – завершении проекта.

Резервное время - количество времени, необходимое для выполнения определенной задачи в рамках проекта, фазы и т.д., которое не вызовет задержки выполнения последующих задач.

Критическая деятельность - задачи проекта на выполнение которых не отпущено резервного времени.

Время ввода - время, в ходе которого предшествующее мероприятие должно быть целиком завершено.

Время запаздывания - ранний срок, к которому должна быть выполнена следующая задача.

Резерв времени - резерв времени и ресурсов необходимый для выполнения проекта в целом. Положительный резерв указывает на опережение плана – графика проекта, отрицательный, наоборот, указывает на отставание, нулевой резерв указывает на выполнение проекта в срок[29].

Шаги 3.1-3.4, в отличие от оценки методом PERT, модифицированы формулой [A.2], которая позволяет учесть не только мнение руководителя проекта, но и суммарное мнение экспертов.

4.Начало работ по проекту

4.1. Детализировать активную фазу проекта FA_m на элементарные пакеты работ P_i ;

4.2. Для каждого пакета работ P_i фазы m оценить длительности $\hat{t}_{m,P_i,k}$ для каждого эксперта;

4.3. Для первого завершённого пакета работ (все работы внутри пакеты закончены) активной фазы FA_m рассчитать коэффициент $K=F/A$ (отношение прогнозного значения оценок эксперта к фактической длительности);

4.4. Построить график зависимости K от процента продвижения пакета работ.

4.5. Для каждого эксперта k в пакете работ P_i фазы m , рассчитать $EQF_{m,P_i,k}$;

4.6. На основе полученных данных, переопределить значение весовых коэффициентов доверия экспертов $\alpha_{m,P_i,k}$ для пакета работ P_{i+1} фазы m как

отношение
$$\alpha_{m,P_{i+1},k} = \frac{EQF_{m,P_i,k}}{\sum_{k=1}^{V_{m,i}} EQF_{m,P_i,k}} ;$$

4.7. На основе полученных коэффициентов $\alpha_{m,P_{i+1},k}$, определить оценку длительности критического пути проекта по формуле [А.3];

4.8. Разместить полученную информацию в базу экспертных оценок;

Для каждого завершённого пакета работ P_i фазы m сохранить параметры для

базы знаний: $Y_{i,k} = (V_k, \alpha_{i,k}), \alpha_{m,P_i,k}, EQF_{m,P_i,k}$;

5. Повторить шаги с 4.2 по 4.8 для всех оставшихся фаз.

Стоимость и Функциональность могут быть оценены как функции времени.

Теорема. Алгоритм ДОПС оценивания времени цикла разработки по методу ДО конечен.



1. Исходные ограничения.

1.1. *Определенный* программный проект имеет чётко обозначенное начало и конец и ограничен во времени, таким образом, разбиение проекта на фазы и количество фаз проекта конечно.

1.2. Возможен переход из пассивной фазы проекта в активную фазу, однако обратный переход исключен.

1.3. В активной фазе *определенный* программный проект не может находиться бесконечно, так как нарушение этого требования переводит организацию на более низкий уровень зрелости, что противоречит условиям применения алгоритма. Таким образом, время выполнения проекта всегда конечно.

2. Представим предложенный алгоритм в виде укрупнённой блок-схемы ВД.



Рис. 36 Укрупнённая блок-схема алгоритма ВД

Объединим все действия в три крупных:

R_1 = Оценка длительности фаз проекта;

R_2 = Анализ наличия активных фаз (Есть ли активные фазы проекта). Данный шаг алгоритма конечен в силу исходных ограничений, обозначенных в п.1.;

R_3 = Конец проекта/

3. Представим укрупненную блок-схему ВД в виде сети Петри, так как вырожденным случаем параллельной системы обработки будет система с единственным процессом, который описывается программой. Структуру управления программой всегда можно представить в виде определенной блок-схемы. Данная блок-схема ВД может быть представлена сетью Петри и соответственно программа может быть представлена соответствующей сетью. Суть перевода блок-схемы ВД в сеть - изменение узлов блок - схемы ВД на переходы сети, дуг блок-схемы ВД на позиции сети. Алгоритм ДОПС метода ДО однозначно описывается блок-схемой ВД. Тогда ВД может быть представлена сетью Петри С.

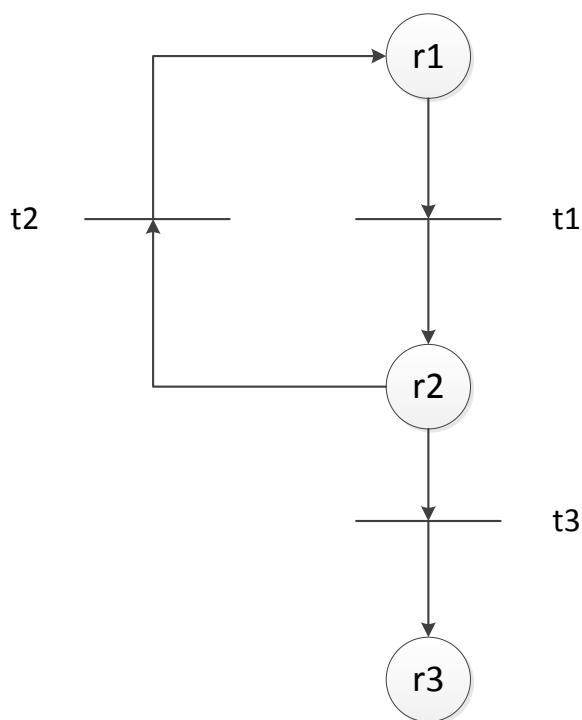


Рис. 37 Сеть Петри С

4. Введём сеть Петри $C = (R, T, I, O)$

$T = \{t_1, t_2, \dots, t_m\}$ - конечное множество переходов, $m \geq 0$.

$R = \{r_1, r_2, \dots, r_n\}$ - конечное множество позиций, $n \geq 0$.

$$R \cap T = \emptyset$$

$I: T \rightarrow R^{\circ}$ - входная функция (отображение из переходов в комплекты позиций).

$O: T \rightarrow R^{\circ}$ - выходная функция (отображение из переходов в комплекты позиций).

3. Маркировка m сети Петри $C = (R, T, I, O)$ - функция, которая отображает множество позиций R во множество N (неотрицательных чисел), $m: R \rightarrow N$.

Следовательно маркированная сеть $M = (C, m)$ представляет собой совокупность структуры сети $C = (R, T, I, O)$ и маркировки m . Таким образом данная сеть может быть записана в виде $M = (R, T, I, O, m)$.

5. Построим для Маркированной сети Петри $M = (C, m)$ дерево достижимости TG, которое представляет множество достижимости сети Петри M , используя стандартный алгоритм.

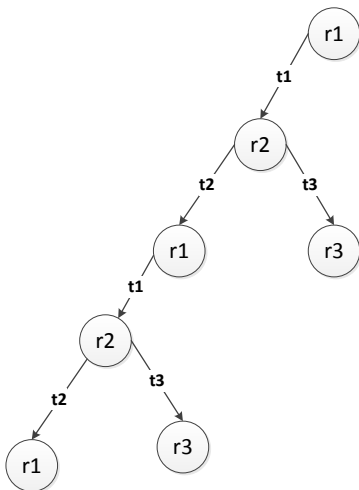


Рис. 38 Дерево достижимости

б. Тогда в дереве достижимости, согласно начальным условиям, будет отсутствовать бесконечное число маркировок, тем самым сеть Петри будет ограничена. Следовательно, алгоритм модели оценивания конечен. ▼

2.7. Пример шага алгоритма оценивания

Приведем пример работы одного шага алгоритма оценивания.

Перестроение $\alpha_{k,i}$ на основе пакета работ j для $j+1$ пакета работ.

Таблица 32. Исходные данные

Пакет работ 1	Продвижение пакета работ проекта (%)	Прогнозируемое время разработки	Коэффициент $K=F/A$ (Forecast/Actual)
Эксперт А	0	30	1.5 =(20/30)
	20	16	0.8
	50	26	1.3
	65	24	1.2
	100	Реальная длительность =20	EQF= 3.6
Эксперт В	0	40	2.0
	20	36	1.8
	50	32	1.6
	65	30	1.5
	100	Реальная длительность =20	EQF= 1.42
Эксперт С	0	28	1.4
	20	26	1.3
	50	24	1.2
	65	22	1.1
	100	Реальная длительность =20	EQF= 4.26

Вывод: Эксперт С имеет лучшую оценку в данной области, среднюю оценку В, наихудшую А.

На основе полученных данных переопределяем значение весовых коэффициентов доверия экспертов $\alpha_{k,i}$ по формуле (F.25), для следующего пакета работ данной фазы как отношение:

$$\alpha_{k,i} = \frac{EQF_{k,i}}{\sum_{k=1}^n EQF_{k,i}} \quad (\text{F.25})$$

$$\sum_{k=1}^3 EQF_{k,i} = 3,6 + 1,42 + 4,26 = 9,28$$

$$\alpha_{1,i} = \frac{EQF_{k,i}}{\sum_{k=1}^n EQF_{k,i}} = \frac{3,6}{9,28} \approx 0,388$$

$$\alpha_{2,i} = \frac{EQF_{k,i}}{\sum_{k=1}^n EQF_{k,i}} = \frac{1,42}{9,28} \approx 0,153$$

$$\alpha_{3,i} = \frac{EQF_{k,i}}{\sum_{k=1}^n EQF_{k,i}} = \frac{4,26}{9,28} \approx 0,459$$

Таблица 33. Новые значения коэффициентов доверия экспертов

Эксперт	$\alpha_{k,i}$ для пакета работ j	$\alpha_{k,i}$ для пакета работ j+1
Эксперт А	0.3(3)	0.388
Эксперт В	0.3(3)	0.153
Эксперт С	0.3(3)	0.459

2.8. Программная реализация предложенного алгоритма

Существующие на рынке программное обеспечение в области управления проектами имеет в основном схожие характеристики. К ним можно отнести: построение ИСР, распределение ресурсов по задачам, контроль прогресса по проекту, анализ объёмов необходимых работ, создание последовательности работ критического пути. На рынке однопользовательских решений продукт Microsoft Project является монополистом, имея 80 %. По отчетам Gartner на рынке корпоративных систем решения от Microsoft и Oracle Primavera занимают 99% рынка. Учитывая распространённость Microsoft Project, возьмём его в качестве базового программного обеспечения для предложенной СППР.

На основании полученного алгоритма была реализована надстройка для MS Project, которая позволяет:

- ✓ поддержать независимую работу экспертов по проектам;
- ✓ автоматизировать процесс сбора и анализа экспертных оценок;
- ✓ агрегировать данные, получаемые от экспертов;
- ✓ строить конус неопределённости для каждого эксперта внутри фазы проекта;
- ✓ переопределять значения весовых коэффициентов для каждого эксперта;
- ✓ накапливать данные для базы знаний;
- ✓ возвращать данные в MS Project и производить оценку длительности критического пути проекта;
- ✓ отображать список всех доступных проектов;
- ✓ содержит краткое описание каждого проекта;
- ✓ состав участников команды проекта с их текущей нагрузкой;
- ✓ позволяет руководителю проекта отслеживать ход работы над проектом;
- ✓ обеспечивает ролевой доступ.

Выводы по второй главе

Согласно методу PERT средняя продолжительность проекта есть сумма средних показателей времени, рассчитанная для операций находящихся на критическом пути проекта, все операции, в том числе стоящие на критическом пути, согласно разработчикам метода PERT, имеют бета-распределение. Todd Little, из компании Landmark Graphics при исследовании более ста коммерческих проектов заметил, что: “I wanted to quantify some of the conventional wisdom about uncertainty. My company had collected metrics for three years on more than 100 commercial software projects, and I saw the opportunity to mine that data to expose trends that I could compare to other industry data. My findings led me to question aspects of that conventional wisdom.” в статье “Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty” получил что: “Each of these contributed somewhat to both the distribution’s variation and the skew toward p10. With these characteristics in mind, some general interpretations of the data and the distributions are possible. The data clearly indicates that Landmark’s software project duration estimation accuracy followed a lognormal distribution with an uncertainty range between p10 and p90 of roughly a factor of 3 to 4. “ The PERT model uses a beta distribution, which has the flexibility to resemble a lognormal distribution. The use of the beta distribution appears to have more to do with its flexible characteristics rather than observed activity. Nonetheless, estimates of espoused high confidence of +/- 2 months, or +/- 20 percent, are still common. Ranges that are given as +/- a constant time or constant percent are missing the problem’s exponential nature [16]. Исходя из результатов полученных автором и косвенных подтверждений из мировой практики в области управления программными проектами можно сделать следующие выводы - оценки экспертов, проводимые в течение проекта:

- имеют различный вес, который необходимо учитывать;

➤ имеют распределение более близкое по своей структуре к логнормальному распределению.

2. Предложенный метод динамической оценки учитывает веса экспертов внутри каждой фазы проекта и тот факт, что оценки экспертов имеют распределение более близкое по своей структуре к логнормальному распределению. Исходя из этого для текущего момента времени t имеем более точную оценку работ, предстоящих в ближайшей фазе. Это достигается за счет использования экспертных оценок и перестроения весов в зависимости от качества прогноза эксперта.

Глава 3. Использование предложенного метода в реальных проектах

3.1. Проект 1.

Создание комплекса полунатурного моделирования интегрированных систем управления воздушным движением. В течение многих лет в ФГУП «ГосНИИАС» создавался программный комплекс имитационного математического моделирования процессов организации и управления воздушным движением - Комплекс имитационного моделирования организации воздушного движения (КИМ ОрВД) для воздушного пространства РФ [74]. Возможности комплекса позволяют на сегодняшний день проводить различные исследования по анализу эффективности использования воздушного пространства РФ и обеспечивать опережающий анализ различных организационных и технических решений, связанных с реорганизацией принципов планирования и управления воздушным движением. Однако отсутствие бортовой компоненты в КИМ ОрВД не позволило полностью отработать и проводить исследования, связанные с функциональным взаимодействием пилотов и бортовой авионики, наземной компоненты и диспетчеров обеспечения воздушного движения (ОВД) и планирования, автоматизированных систем управления воздушным движением (УВД), как в существующих условиях выполнения полетов, так и перспективных. В рамках НИР «Разработка прототипов и комплектующих интегрированной модульной авионики (ИМА) – систем бортового радиоэлектронного оборудования (БРЭО) и агрегатов самолетов и вертолетов Гражданской Авиации (ГА) и технологий их создания», в ГосНИИАС был создан комплекс полунатурного моделирования интегрированных систем управления воздушным движением (КИС УВД) [75,76].

Целями создания КИС УВД являются:

- отработка и исследования функционального взаимодействия пилотов и бортовой Авионики, диспетчеров и средств автоматизации управления воздушным движением (УВД) при решении задач наблюдения и самолетовождения в сложных условиях;
- отработка перспективных функциональных возможностей борта в части наблюдения и самолетовождения, связанных с делегированием ответственности на борт;
- оценка эффективности применения новых бортовых средств и возможностей CNS/ATM (Communication Navigation Surveillance/Air Traffic Management);

С формальной точки зрения такой комплекс является сложной технической системой управления сверхвысокой размерности, включающей в себя различные программно-реализуемые имитационные модели. Объем математических моделей определяется исследовательскими системными функциональными задачами, которые предлагается решать с использованием комплекса. Для проверки частных аппаратурных решений (например, подтверждения нормального взаимодействия двух-трех приемо-передающих устройств между собой) предусматривается замена части имитационных моделей их реальными прототипами и их сопряжение с программно-реализуемой частью моделирующего комплекса (например, была произведена замена имитационной модели диспетчера ОВД на АРМ диспетчера ОВД, внедрены каналы диспетчерской связи и т.д.) В состав КИС УВД включены имитационные модели наземной и бортовой компонент.

В состав наземной компоненты комплекса входят:

- модель внешней обстановки. Предназначена для имитации движения центра масс воздушных судов (ВС), управления движением ВС от взлета

до посадки и поддержка моделирования в соответствии с концепцией “GATE TO GATE”;

- модель метеонаблюдений. Предназначена для имитации различных опасных метеоявлений в динамике их развития;
- модель автоматизированной системы управления воздушным движением (АС УВД). Предназначена для имитации действий диспетчеров АС УВД во всем объеме моделируемого воздушного пространства в части взаимодействия модели с бортом;
- модель централизованного планирования. Предназначена для имитации функций планирования и регулирования потоков ВД;
- модель планирования и управления в районе аэродрома – AMAN /DMAN (Arrival Manager / Departure Manager). Предназначена для имитации функций планирования и регулирования потоков воздушного движения (ВД) в районе аэродрома по прилету/вылету;
- модель наземного комплекса наблюдения. Позволяет проводить моделирование радиолокационных средств обзора и сопровождения ВС, имитировать работу альтернативных средств наблюдения;
- модель «эфира». Позволяет проводить имитацию использования эфира в пространстве и времени при условии одновременной работы всех абонентов, как по цифровым линиям связи, так и по голосовым;
- модель наземной линии связи. Предназначена для имитации обмена сообщениями абонентов по сети интернет и телеграфным линиям.

В состав бортовой компоненты («Кабина») входят:

- модель бортового комплекса связи. Предназначена для имитации приема/передачи сообщений на информационном уровне;
- модель бортовой аппаратуры АЗН-В. Предназначена для имитации информационного взаимодействия борта с системой связи и функциональным ПО комплекса;

- модель движения ВС. Ключевым элементом «Кабины» является математическая модель уравнений движения ВС под управлением пилота;
- модель Flight Management System (FMS). Макет ПО системы самолетовождения;
- модель программного обеспечения перспективных функциональных приложений функций наблюдения и самолетовождения.

Для управления экспериментом и организации распределенного моделирования в состав комплекса включены:

АРМ управления экспериментом.

Данная модель предназначена для:

- формирования сценария моделирования:
 - определение конфигурации эксперимента (подключаемые АРМ и макеты);
 - формирование потоков воздушного движения и структуры воздушного пространства;
 - задание параметров функционирования комплекса (модельное время, масштаб моделирования, параметры конфигурации системы).
- организации распределённого моделирования;
- средства анализа ситуации в процессе моделирования и пост полётный анализ.

Комплекс синхронизации времени

Данная модель обеспечивает определение, хранение и выдача значений текущего времени, синхронизированного со шкалой всемирного скоординированного времени UTC.

Функциональная схема комплекса:

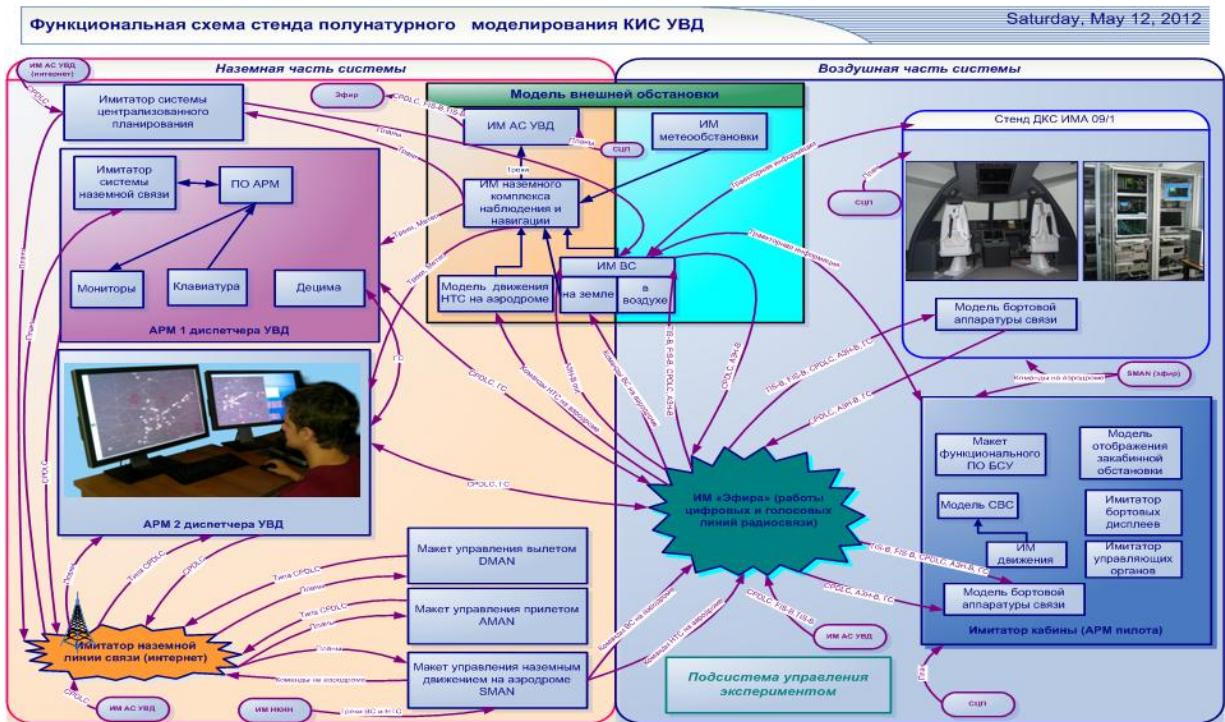


Рис. 39 Функциональная схема комплекса КИС УВД

Комплекс КИС УВД разрабатывается как универсальное средство оценки эффективности и особенностей выполнения полетов в любых задаваемых условиях, как со стороны системы УВД, так и с точки зрения бортовой компоненты. Комплекс в настоящее время, например, позволяет отрабатывать различные функции и отдельные процедуры бортовой системы обеспечения эшелонирования ASAS (Airborne Separation Assurance System) [76]; перспективные функциональные приложения функций наблюдения и самолетовождения; функции наземного управления движением на аэродроме; различные алгоритмы управления и планирования потоков воздушного движения.

3.2. Проект 2.

Создание комплекса имитационного моделирования движения воздушных судов и наземных транспортных средств на поверхности аэродрома. Современный аэропорт представляет собой сложную систему,

правление которой является очень непростой и ответственной задачей. Цена допущенной ошибки здесь весьма велика. Движение на территории аэродрома включает в себя перемещение воздушных судов (ВС) и наземных транспортных средств (НТС) (например, снегоочистители, тягачи, транспортные средства для перевозки персонала, багажные тележки) на таких участках как места стоянки, перроны, рулежные дорожки и взлетно-посадочные полосы (ВПП). В настоящее время экипажи осуществляют перемещения по территории аэродрома с использованием бумажных карт и визуальных средств аэродрома. Визуальные средства обзора из кабины включают в себя осевую разметку, боковую разметку, световую сигнализацию, указатели, другие воздушные суда, другие транспортные средства, различные местные предметы, здания, рулежные дорожки, взлетно-посадочные полосы и т.п. Выполнение движения по маршруту предполагает местное управление воздушным судном, включающее маневрирование воздушным судном на маршруте (выбор наилучшей скорости и выдерживание осевой линии). Кроме того, экипаж воздушного судна имеет задачу поддержания общей информированности, которое предполагает контроль местоположения относительно точки назначения и опасных мест, возникающих опасностей и поворотов. Территория аэродрома является сложной и весьма оживленной средой, ставящей перед экипажем множество задач (например, навигация, выполнение карт проверок, радиообмен с оперативным персоналом компании), кроме того, она может быть незнакома экипажу. К этому следует добавить не совсем подходящие метеоусловия или условия освещения. Для автоматизации процесса управления движением на поверхности аэродрома активно разрабатываются и внедряются в эксплуатацию усовершенствованные системы контроля и управления наземным движением. Основным предназначением такой системы является предупреждение пользователей о конфликтных ситуациях (сближение, столкновение, выкатывание ВС и НТС на ВПП, столкновения с

искусственными препятствиями). В настоящее время диспетчеры наземного движения получают информацию о местоположении движущихся объектов при помощи средств визуального контроля и в некоторых случаях с использованием радиолокатора контроля за наземным движением и приемников сигналов от бортовых ответчиков. После приема и обработки принятые данные о местоположении движущихся объектов отображаются на рабочем месте диспетчера. Полученная информация о движении на территории аэродрома присутствует только на рабочем месте диспетчера руления, а участники движения от нее полностью изолированы. Выходом из проблемы может служить подход, предлагаемый западной Европой и Америкой в научно-исследовательских программах модернизации систем ОпВД: Single European Sky ATM R&D (SESAR), Next Generation Air Traffic Service (NextGen) [77,78] – оснащение участников движения приемопередатчиками информации автоматического зависимого наблюдения в режиме радиовещания (АЗН–В). В таком случае каждый движущийся объект сможет увидеть на бортовых средствах отображения информацию о наземном движении других участников, оборудованных передатчиками. Также каждый движущийся объект должен быть обеспечен идентичной электронной картой аэродрома (движущейся картой местности) – Surface Moving Map (SMM). Электронная карта аэродрома может быть построена по данным аэродромной картографической базы данных – Aerodrome Mapping Database (AMDB) [79,80]

Целями комплекса являются:

- имитация наземного движения ВС и НТС для отработки перспективных функций системы управления наземным движением и контролем;
- отработка и исследования взаимодействия пилотов и бортовой авионики, наземных служб и диспетчеров, средств автоматизации управления движением на поверхности аэродрома при решении задач наблюдения и самолётовождения в сложных условиях;

- создание электронной карты аэродрома на основе аэродромной картографической базы данных.

Имитационная модель движения ВС и НТС на поверхности аэродрома должна обеспечивать достаточно подробную имитацию процессов происходящих на территории аэродрома для ВС и НТС.

Для ВС имитируются следующие процессы:

- посадки;
- движения по ВПП;
- движения по рулежной сети аэродрома;
- стоянки: разгрузки; смена рейса; заправки; технического обслуживания;
- деайсинга;
- разворота;
- взлета.

Процессы для НТС имитируются аналогично.

В состав имитационной модели движения ВС и НТС на поверхности аэродрома входят:

- модель движения ВС и НТС – Предназначена для имитации движения центра масс ВС и НТС, управления движением ВС и НТС на аэродроме. Движение ВС имитируется от посадки до взлета (поддержка моделирования в соответствии с концепцией “GATE TO GATE”);
- модель наблюдения за движущимися объектами – Предназначена для имитации аэродромной системы наблюдения и моделирования переходных ситуаций, когда одни ВС или НТС оборудованы АЗН–В, а другие нет;
- наземная система отображения – Предназначена для имитации системы отображения места работы диспетчера наземного движения;
- модель базы данных аэродрома;
- модель формирования базы данных маршрутов ВС и НТС;

➤ модель задания движения ВС и НТС;

На вход в модель подается поток ВС, прибывающих в данный аэродром, а на выходе формируется поток вылетающих ВС.

Функциональная схема комплекса:

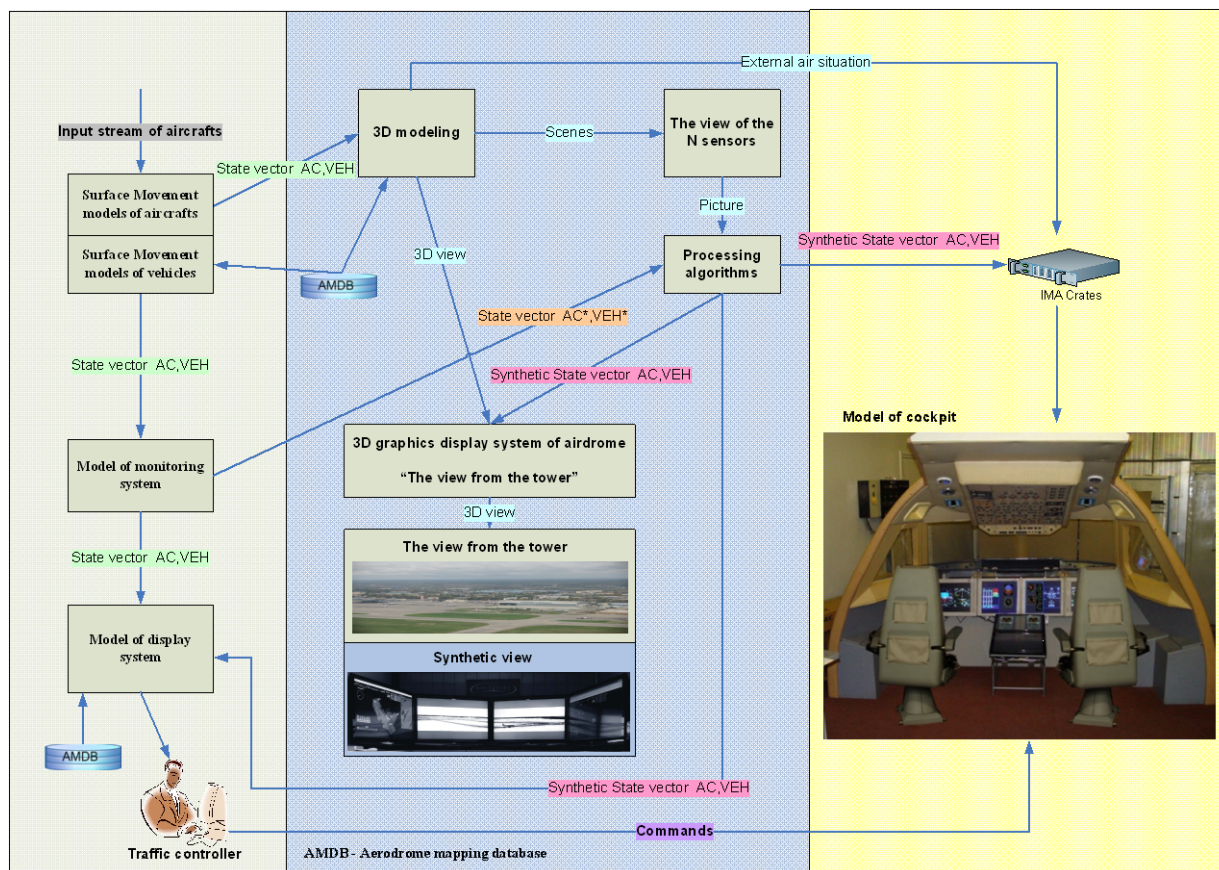


Рис. 40 Функциональная схема комплекса

Представленные комплексы являются сложными техническими системами управления сверхвысокой размерности, включающие различные программно-реализуемые имитационные модели. Их уникальность заключается в интеграции имитационных моделей с макетами автоматизированных рабочих мест и аппаратурой, а также в практическом отсутствии подобных отечественных универсальных комплексов, предназначенных для решения разноплановых исследовательских задач.

3.3. Проект 3.

Внедрение автоматизированного решения customer value management (CVM). CVM – проект первого внедрения в РФ автоматизированного решения для моделирования и управления кампаниями на основе ПО IBM Unica и SPSS.

Цели проекта внедрения:

- реализация основного и дополнительного контура системы;
- построение витрины данных ПО IBM Unica и SPSS на основе корпоративного информационного хранилища;
- настройка и интеграция ПО IBM Unica и SPSS с витриной данных;
- реализации исторических снимков атрибутов;
- настройка и интеграция с внешними системами и каналами коммуникации;
- создание витрины для построения отчётов.

Функциональная схема:

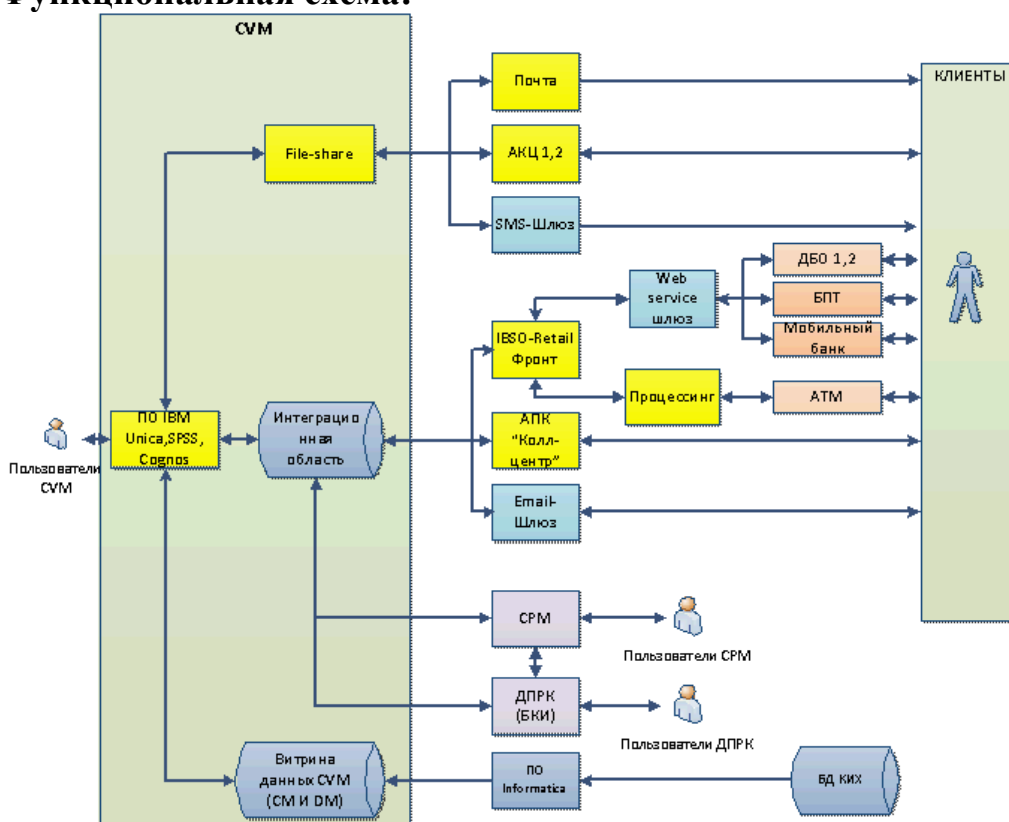


Рис. 40 Функциональная схема CVM

3.4. Проект 4

Разработка и внедрение системы дистанционного банковского обслуживания в российском банке.

Цели проекта:

- разработка интерфейса собственной системы дистанционного банковского обслуживания (ДБО) клиентов;
- интеграция системы с внутренними учётными комплексами банка;
- разработка хранилища данных;
- реализации исторических снимков атрибутов;
- настройка и интеграция с внешними системами и каналами коммуникации;
- создание витрины для построения отчётов.

3.5. Проект 5

Разработка и внедрение нового канала доступа к системе дистанционного банковского обслуживания - “мобильный банк” для устройств на операционных системах iOS, Android, Windows Phone.

Цели проекта:

- разработка интерфейса системы для различных мобильных операционных систем;
- интеграция системы с системой ДБО и внутренними учётными комплексами банка;
- настройка и интеграция с внешними системами и каналами коммуникации;

3.6. Проект 6

Внедрение специализированного модуля оптимизации “Contact Optimization” – проект первого внедрения в РФ автоматизированного

решения для математической оптимизации и управления коммуникациями на основе ПО IBM Contact Optimization.

Цели проекта внедрения:

- построение витрины данных ПО IBM Contact Optimization на основе корпоративного информационного хранилища;
- разработка предсказательных модели в связке клиент-канал-предложение;
- настройка и интеграция модулей ПО IBM: Unica, SPSS, Contact Optimization;
- настройка и интеграция с внешними системами и каналами коммуникации;
- создание витрины для построения отчётов.

3.6. Пример работы алгоритма ДОПС для третьего проекта

Данный проект был признан успешным, завершён раньше объявленного срока с достижением всех заявленных первоначально функций. На этапе планирования заявленная длительность проекта согласно методу PERT соответствовала 499 рабочим дням, фактическая длительность составила 427 дня. Были задействованы 3 независимых эксперта, которые провели более 100 оценок. На их основании рассчитывалась длительность критического пути проекта с шагом 5%.

Таблица 34 Результаты работы алгоритма для проекта SVM

A	B	C	D	F	G	H	I	G	K	L	P
100										427	0,00%
95	0,1696	0,2183	0,1910	0,2930	0,3771	0,3300	418	401	448	420	-1,69%
90	0,1600	0,2058	0,1474	0,3119	0,4010	0,2872	369	420	471	426	-0,34%
85	0,1813	0,2036	0,2504	0,2854	0,3205	0,3941	388	441	477	437	2,36%
80	0,2125	0,1856	0,2431	0,3313	0,2895	0,3791	402	461	359	403	-5,55%
75	0,1884	0,1747	0,2218	0,3221	0,2987	0,3792	507	503	393	463	8,51%
70	0,1995	0,1519	0,2096	0,3556	0,2709	0,3736	425	584	411	461	7,89%
65	0,2028	0,1404	0,2027	0,3715	0,2571	0,3713	372	623	420	451	5,70%
60	0,2053	0,1313	0,1998	0,3827	0,2448	0,3725	288	655	425	424	-0,72%
55	0,1994	0,1164	0,1935	0,3915	0,2285	0,3800	246	790	441	418	-2,13%
50	0,2352	0,0969	0,1894	0,4510	0,1858	0,3632	246	895	450	434	1,54%
45	0,2325	0,0881	0,1809	0,4636	0,1756	0,3608	204	1008	491	416	-2,69%
40	0,2740	0,0770	0,1631	0,5329	0,1498	0,3173	161	1107	537	414	-2,99%
35	0,2734	0,0730	0,1519	0,5486	0,1465	0,3048	161	1228	575	446	4,56%
30	0,2449	0,0665	0,1399	0,5427	0,1473	0,3100	159	1283	610	474	11,12%
25	0,2139	0,0608	0,1350	0,5220	0,1485	0,3295	116	1427	663	536	25,50%
20	0,1568	0,0569	0,1189	0,4715	0,1712	0,3573	117	1198	753	536	25,52%
15	0,1547	0,0639	0,1039	0,4798	0,1980	0,3222	112	1492	889	628	46,99%
10	0,1314	0,0551	0,0769	0,4988	0,2093	0,2919	108	1305	1134	841	96,84%
5	0,1608	0,0681	0,0696	0,3300	0,3300	0,3300	115	918	1361	790	85,02%

Здесь в столбце А процент продвижения работ, находящихся на критическом пути проекта; В, С, D содержат значения EQF экспертов; F, G, H содержат значения коэффициентов доверия экспертов; I,G,K – прогнозы экспертов; L – расчёт длительности согласно методу; P – относительное отклонение длительности критического пути.

Выводы по третьей главе

Сформулируем основные результаты использования метода:

- эксперты осуществляют оценку неравномерно, алгоритм перераспределяет значения весовых коэффициентов доверия экспертов в зависимости от качества оценки на предыдущем этапе;
- на этапе выполнения 35%-40% работ, находящихся на критическом пути, на основе предложенного алгоритма можно было сделать вывод о первоначально завышенных сроках выполнения проекта;

- использование метода позволило правильно оценивать эксперта и команду в целом, что дало возможность принимать верные управленческие решения;
- благодаря повышению точности оценки на 18,80% (в отличие от метода PERT), была достигнута экономия в 72 рабочих дня, это позволило достичь экономии более 22 миллионов рублей, что составляет порядка 8% стоимости всего проекта;
- на основе данного проекта сформирована база экспертных оценок, которую можно использовать при повторном внедрении данного ПО или реализации схожих проектов в качестве реального эксперта или для повышения точности оценки.

Приведём результаты использования метода в описанных выше проектах (данные в рабочих днях) (табл.35)

Таблица 35 Результаты использования метода

Название проекта	Длительность проекта согласно методу PERT	Метод ДО	Фактическая длительность	Повышение точности оценки
Проект 1	217	247	256	12,14%
Проект 2	154	181	198	14,91%
Проект 3	499	420	427	18,80%
Проект 4	743	829	987	10,37%
Проект 5	439	528	571	16,85%
Проект 6	125	147	158	14,96%

Глава 4. Функциональные показатели процессов разработки ПО

Для успешного выполнения ИТ - проекта недостаточно выбрать эффективные технологии, методы и инструменты разработки, найти бюджет и квалифицированных разработчиков. В любой организации существует ряд

правил и методики, объединённых в единую методологию, согласно которой участники рабочей группы проекта распределяют задачи, взаимодействуют друг с другом, создают проектные документы и исходный код. Данные правила должны быть чётко закреплены и организованы и формально задокументированы, совокупность именно данных правил определяет процесс разработки. Каждые пять-семь лет сложность приложений и их объем постоянно повышаются, так же, как и стоимость разработки. В настоящее время, основная тенденция это не только разработка качественного продукта, но и возврат потраченных инвестиций и получение прибыли. Так как разработка ПО итерационно усложняется во всех аспектах, на рынке сформировалась стойкая потребность давать чёткий ответ на вопрос - “может ли организация вообще разработать требуемый продукт”. В результате возникла острая необходимость в наборе свойств, признаков, критериев, позволяющих количественно оценить степень зрелости предприятия и вероятность его успеха при создании ПО. Функциональные показатели определяют основные свойства, которые характеризуют функциональную пригодность ПО удовлетворять заданные потребности [81].

К основным функциональным показателям принято относить следующие характеристики процесса разработки ПО:

- 1) качество разработки бизнес - требований;
- 2) утверждённая архитектура проекта;
- 3) качество разработки функциональных требований;
- 4) качество программного кода;
- 5) функциональные и архитектурные прототипы;
- 6) глубина и полнота тестирования.

Все перечисленные процессы и характеристики включаются в более общее понятие – методологию разработки ПО. Следование выбранной

методологии в совокупности с уровнем зрелости организации в конечном итоге определяет качество разработки программного обеспечения. К основным методологиям и методам можно отнести следующие, описанные ниже, модели.

4.1 Последовательные модели (Каскадная, водопада, V-образная)

Предложена Роусе W.W. в 1970 году[82]. Основная идея - процесс разработки ПО делится на определённые фазы, выполняемые строго последовательно. Классическая каскадная модель или модель водопада включает в себя следующие области:

- разработка требований;
- анализ и дизайн;
- реализация;
- валидация и тестирование;
- развёртывание системы.

Последовательным развитием каскадной и водопадной моделей явилась **V-образная модель** (рис.40).

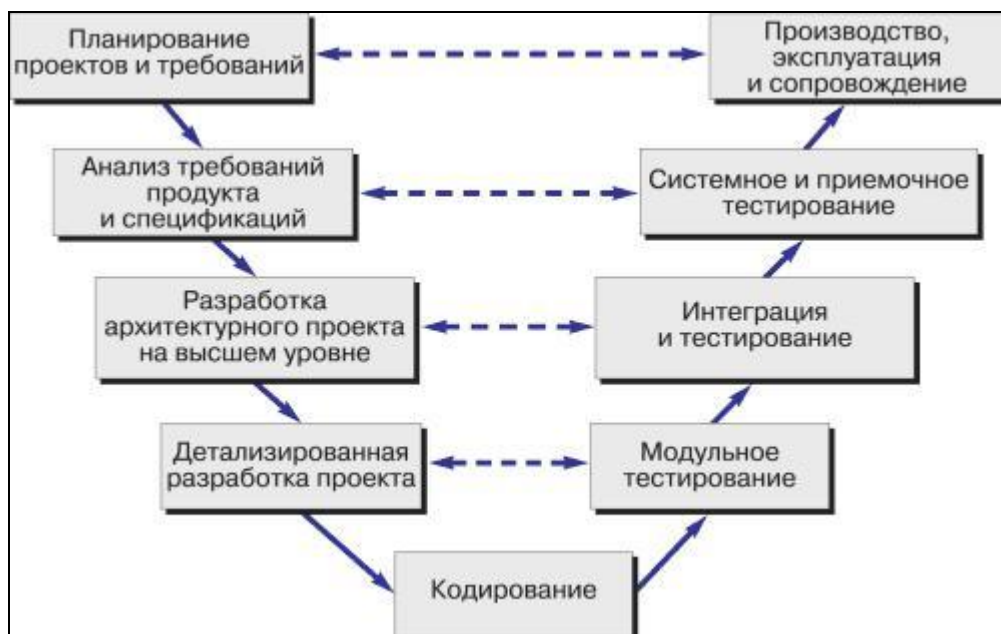


Рис. 40. V-образная модель.

Применение V-образной модели позволила существенно повысить качество разработки ПО за счёт ориентации на тестирование, позволив решить задачу соответствия созданного продукта выдвигаемым требованиям, за счет процедур верификации и аттестации на ранних этапах разработки. По своей сути, V-образная модель – модификация каскадной, обладает многими её недостатками, например, слабо приспособлена к возможным изменениям требований заказчика.

Предложенный в диссертационной работе метод динамической оценки может быть применён при использовании последовательных моделей, однако большого преимущества по сравнению с классическим подходом он не принесёт. Это связано со следующим фактами:

- последовательные модели хорошо работают в проектах, где требования могут быть заранее чётко определены и зафиксированы. Метод динамической оценки фокусируется на том факте, что требования постоянно изменяются в ходе выполнения проекта;
- использование предложенного метода позволит повысить первоначальную точность оценки, благодаря использованию агрегированного мнения экспертов о фазах проекта, но стоит заметить, что дальнейшее последовательное уточнение точности оценок не произойдёт.

4.2 Спиральная модель

Спиральная модель, предложенная Boehm B.W. в 1988 г., объединила итерационный процесс проектирования и классический каскадный подход на основе идеи создания прототипов [82] (рис. 41).

4.3 Итеративные модели

Модель итеративной (инкрементальной) разработки - эволюционное развитие каскадной модели. Процесс разработки состоит из последовательности повторяющихся итераций, каждая из которых является полноценным проектом с определёнными фазами. В результате каждой итерации продукт получает улучшения своей функциональности или новую компоненту. После завершения последней итерации проект получает полный набор функции и требований, определенный границами проекта.

Исходя из требований заказчика, команда проекта выбирает результат итерации:

- ✓ промышленную систему с ограниченными функциями;
- ✓ архитектурные или функциональные или прототипы [83,84].

Во второй половине 1990-х годов в компании Rational Software была создана **итеративная модель RUP**. RUP описывает общий абстрактный процесс, на основе которого команда проекта должна создать специализированный, адаптированный процесс, ориентированный на внутренние потребности[84].

Основные характеристики процессов RUP.

Разработка требований. На основании прецедентов использования описываются требования к системе, полный набор которых образует модель прецедентов. Прецедент использования – описание сценария взаимодействия пользователя с системой, который полностью описывает пользовательскую задачу. Согласно методологии RUP, функциональные требования (ФТ) должны быть представлены в виде прецедентов использования.

Итеративная разработка. В методологии RUP проект состоит из последовательности итераций с продолжительностью от двух до шести недель. Прецедент использования - основная единица планирования

итераций. Перед началом итерации определяется список прецедентов использования, для реализации в данной итерации. В ходе реализации проекта в итеративной модели, могут быть внесены необходимые изменения в требования, архитектуру, реализацию и проектные решения.

Архитектура RUP. RUP - ориентированная на архитектуру методология. В данной методологии на ранних этапах проекта начинаться реализация и тестирование архитектуры системы. В RUP используется понятие исполняемой архитектуры, которая позволяет в первую очередь реализовать архитектурно значимые прецеденты.

Microsoft Solutions Framework (MSF) является гибкой методологией, построенной на итеративной модели разработки, опирается на практический опыт и описывает управление рабочими процессами и людьми в процессе разработки. Основная особенность MSF - создание эффективной, не бюрократизированной проектной команды. Для этого MSF предлагает нетрадиционные подходы к распределению ответственности, организационной структуре и принципам взаимодействия внутри команды[88].

Предложенный метод динамической оценки даёт наилучшие результаты при использовании итеративных моделей, по сравнению с другими. Это связано со следующим фактами:

- процесс в итеративных моделях состоит из серии повторяющихся итераций, которые наилучшим образом ложатся на предложенный алгоритм;
- каждая из фаз является полноценным мини-проектом, что в совокупности с использованием предложенного метода даёт последовательное повышение точности оценки;
- реализация основных (существенно важных) функций может быть завершена в ходе первых итераций. В случае изменения требований со

стороны заказчика предложенный метод позволит повысить точность последующих прогнозов за счёт накопления знаний на предыдущих фазах. Необходимые изменения и дополнения могут быть реализованы в последующих итерациях;

➤ в итеративных моделях основные проектные риски находят своё решение на первых итерациях, использование предложенного метода позволяет существенно понизить проектные риски благодаря возможности гибкого учёта мнения экспертов;

➤ итеративные модели обладают широкими возможностями настройки и могут быть настроены под специфику конкретной организации и проекта.

4.4. Гибкие методологии

В начале 2000-х годов группой экспертов по легковесным методологиям были сформулированы принципы гибкой разработки ПО - Agile Manifesto[87].

Общие особенности гибких методологий:

➤ ориентированность на разработчиков и заказчиков. Основная идея собрать в проектной команде профессионалов, которые определяют успешность проекта в большей степени, чем процессы и технологии;

➤ вместо формальных спецификаций использование устных обсуждений, которые служат главным способом коммуникации внутри проектной команды;

➤ итеративная разработка с минимальной длительностью итерации. В результате каждой итерации происходит выпуск полноценной работающей версии продукта;

➤ В начале проекта команда не пытается зафиксировать требования и затем следовать жёстко определённым плану. Изменения могут быть сделаны на любом этапе проекта.

Экстремальное программирование (eXtreme Programming - XP) - гибкая методология разработки ПО созданная Бекем К. в 1996 г. в результате работы над проектом в компании Chrysler [85]. Методология наследует общие принципы гибких методологий, достигая при их помощи инженерных практик:

- в команде проекта постоянно присутствует заказчик, обладающий детальной информацией о необходимой функциональности, который определяет приоритеты ФТ, задач и оценивает качество создаваемой системы;
- пользовательские истории и приёмочные тесты являются средством спецификации требований, по своей сути являясь короткими неформальными описаниями прецедентов использования системы;
- весь создаваемый код автоматически покрывается юнит-тестами;
- максимально простая архитектура системы. Методология XP не рекомендует проектировать в расчёте на будущее системы. Текущая архитектура должна поддерживать существующую функциональность;
- изменение архитектуры требует постоянной переработки кода. XP поощряет коллективное владение кодом;
- сделанные разработчиками изменения, после автоматического тестирования попадают в репозиторий, этап интеграции отсутствует;
- парное программирование и сорокачасовая рабочая неделя (в долгосрочной перспективе происходит повышение производительности проектной команды за счет уменьшения стресса и переутомления).

Методология SCRUM предоставляет эмпирический подход к разработке ПО. Данный процесс адаптивен и основан на повторяющихся циклах. В его основе лежат короткие ежедневные встречи “Scrum” и циклические тридцатидневные встречи “Sprint”[89].

Основные принципы Scrum:

- постоянное взаимодействие с заказчиком важнее договорных обязательств;
- гибкий и индивидуальный подход важнее строгих процессов и методов;
- следованию плану может быть изменено при необходимости;
- работающее ПО важнее проектной документации.

Предложенный метод динамической оценки не применим при использовании гибких методологий. Это связано со следующим фактами:

- гибкие методологии применимы только в команде опытных, профессиональных разработчиков. Команда не может быть разбита на несколько частей, так как необходимо прямое общение разработчиков, что невозможно в распределённой географически команде; размер эффективной команды ограничен сверху (10-15 человек). Перечисленные пункты противоречат базису, заложенному в предложенный в диссертационной работе метод принципу Wideband Delphi;
- гибкие методологии не делают попыток предотвратить размывание границ проекта с фиксированной ценой. Фактически проекты обладают жёстким графиком, но переменными границами, что неприменимо в рамках предложенного метода;
- практика поддержки простой архитектуры и отсрочка разрешения нетехнологических рисков обычно приводит к тупику в конце проекта. Может оказаться, что для завершения проекта может потребоваться полное

перепроектирование системы, что не соответствует базису предложенного метода.

4.5 Уровень зрелости организации

Уровень зрелости организации определяется согласно модели Capability Maturity Model Integration (CMMI) [85]. Модель CMMI разработана в Software Engineering Institute в 1990-х годах. Она была создана при объединении следующих моделей разработки: Electronic Industries Alliance Interim Standard, CMM for Software и Integrated Product Development Capability Maturity Model [86].

Цель внедрения CMMI – построение инфраструктуры процессов, устанавливающих единый стандарт выполнения и разработки проектов внутри организации. Для каждого проекта существующий стандартный процесс необходимо модифицировать в соответствии со спецификацией. Исходя из формальных метрик измеряется эффективность существующих процессов, данные процессы постоянно изменяются и оптимизируются. Проект может быть CMMI-совместимым с водопадным, итеративным или другим ЖЦ. Модель CMMI представляет набор элементов, процессов, приёмов, методик, на основе которых формируется итоговый процесс. В организациях внедрение CMMI может происходить на непрерывной или ступенчатой основе. Ступенчатое представление формализует требуемую последовательность шагов для внедрения CMMI, когда каждый шаг модели соответствует достижению кампанией определенного уровня зрелости. Непрерывное представление даёт возможность проводить улучшения в отдельных областях и процессах в произвольном порядке. Основой модели CMMI служат процессные области, общие и специальные задачи и практики. Модель CMMI определяет различные процессные области: планирование проекта, управление рисками, разработка требований и т.д. При ступенчатом

представлении процессные области собраны в пять уровней зрелости, в непрерывном представлении модели любая процессная область находится на одном уровне производительности из шести.

Ступенчатое представление модели СММІ дает организациям возможность классифицировать себя по уровням зрелости:

- “начальный” уровень (уровень 1) – на данном уровне организационные процессы формируются спонтанно и хаотично. Часть проектов могут быть успешными, но вероятность их успешного завершения, соответствие плану проекта и бюджету не предсказуемы;
- “управляемый” уровень (уровень 2) – основная задача данного уровня установление для каждого проекта организации стандартных процессов управления требованиями, планирования, наблюдения, контроля и управления конфигурациями. Процессы должны достигать заявленных целей в соответствующих областях, и общих целей уровня. Для каждого проекта на данном уровне необходимо периодически проводить анализ его соответствия установленным процедурам и метрикам, при необходимости корректировать процессы;
- “определённый” уровень (уровень 3) – включает в себя требования предыдущих двух уровней и добавляет множество необходимых обязательных процессных областей (разработка бизнес-требований, тестирование, интеграция, управление рисками, и т.д.). На данном уровне набор стандартных процессов должен существовать для всей организации, а не для конкретного проекта. Процессы для отдельных проектов создаются на основе настройки стандартных процессов предприятия или организации;
- “количественно управляемый” уровень (уровень 4) – требует измерения метрик качества и производительности существующих в организации

процессов. По сравнению с “определённый” уровнем позволяет получить сравнения и предсказания измеряемых характеристик процессов;

➤ “оптимизирующий” уровень (уровень 5) – наивысший уровень развития организации, согласно СММІ. При помощи метрик 4 -го уровня организация последовательно и циклично изменяет свои процессы для улучшения качества и производительности создаваемых продуктов.

Для применения, предложенного в работе метода, в организации:

- ✓ должен существовать единый набор стандартных процессов для всей организации;
- ✓ достигнута высокая точность планирования и эффективность разработки;
- ✓ получена возможность предсказуемости сроков и бюджета выполнения аналогичных проектов;
- ✓ процессы и промежуточные результаты должны быть высоко документированы.

Исходя из перечисленного выше, для применения предложенного метода организация должна находиться на третьем уровне зрелости и выше.

Выводы по четвёртой главе

- ✓ последовательные модели хорошо работают в проектах, где требования могут быть заранее чётко определены и зафиксированы. Предложенный метод динамической оценки фокусируется на том факте, что требования постоянно изменяются в ходе выполнения проекта;
- ✓ использование предложенного метода в последовательных моделях позволит повысить первоначальную точность оценки благодаря использованию агрегированного мнения экспертов о фазах проекта, но стоит заметить, что дальнейшего уточнения точности оценки не произойдёт;

- ✓ предложенный метод даёт последовательное повышение точности оценки при использовании итеративных моделей, которые состоят из серии повторяющихся итераций;
- ✓ использование итеративных моделей позволяет получить физическую реализацию основных функций в ходе нескольких первых итераций. В случае изменения требований со стороны заказчика предложенный метод позволит повысить точность последующих прогнозов за счёт накопления знаний на предыдущих фазах.
- ✓ в итеративных моделях основные проектные риски находят своё решение на первых итерациях, использование предложенного метода позволяет понизить их благодаря возможности гибкого учёта мнения экспертов;
- ✓ гибкие методологии применимы только в команде опытных, профессиональных разработчиков. Команда не может быть разбита на несколько частей, так как необходимо прямое общение разработчиков, что невозможно в распределённой географически команде. Размер эффективной команды ограничен сверху (10-15 человек). Перечисленное противоречит базису, заложенному в предложенный метод – принципу Wideband Delphi.

Согласно проведённому анализу предложенный метод оценки может быть применён для достижения требуемых функциональных показателей разработки ПО в соответствии с приведённой ниже таблицей. В ней по строкам приведены уровни зрелости предприятия, а по столбцам – методологии разработки. Знак “+” означает возможность применения предложенного метода оценки, знак “-” – невозможность.

Таблица 36. Возможность применения метод оценки применительно к методологиям разработки и уровнем зрелости предприятия.

Уровень зрелости	Последовательные модели		Спиральная модель	Итеративные модели		Гибкие методологии	
	Каскадная	V-образная		RUP	MSF	XP	SCRUM
1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	+	+	+	+	+	-	-
4	+	+	+	+	+	-	-
5	+	+	+	+	+	-	-

Метод динамической оценки применим для организаций с высоким уровнем зрелости (3 и выше по шкале СММІ), для крупных программных проектов, использующих последовательные, спиральные и итеративные модели разработки ПО. Наилучшие результаты достигаются при использовании в организации с 4 и 5 уровнем зрелости и использующих итеративную модель RUP.

Заключение

Основным итогом диссертационной работы является создание метода, алгоритма и системы поддержки принятия решений для прогнозирования длительности цикла процессов разработки программных систем, что выразилось в достижении следующих научных и прикладных результатов.

1.Метод динамической оценки, дающий 10% - 15% эффект в повышении точности оценки времени выполнения программных проектов в организациях с высоким уровнем зрелости для крупных ИТ-проектов, использующих итеративные модели разработки, в частности, RUP.

2.Алгоритм ДОПС, реализующий предложенный метод.

3.Методика формирования базы экспертных оценок, позволяющая на основе выполненных проектов получить числовые характеристики оценок

длительности проектного цикла от различных экспертов, сохранить их в аналитической базе данных, которая в дальнейшем может быть использована для выполнения предварительного прогноза для проектов подобного класса.

4. Разработана СППР на основе предложенного алгоритма. Данная СППР позволяет поддержать независимую работу экспертов по проектам и автоматизировать процесс сбора и анализа экспертных оценок: агрегировать данные, получаемые от экспертов; строить конус неопределённости для каждого эксперта внутри фазы проекта; переопределять значения весовых коэффициентов для каждого эксперта; накапливать данные для базы знаний; возвращать данные в MS Project и производить оценку длительности критического пути проекта; отображать список всех доступных проектов; содержит краткое описание каждого проекта; состав участников команды проекта с их текущей нагрузкой; позволяет руководителю проекта отслеживать ход работы над проектом; обеспечивает ролевой доступ.

5. Создана и собрана аналитическая база экспертных оценок на основе пяти крупных успешно завершённых программных проектов. Полученная аналитическая база экспертных оценок используется в качестве основы “виртуального эксперта”.

Используемая литература

- [1] Арчибальд Р. Управление высокотехнологичными программами и проектами // 3-е изд., переработанное и дополненное - М.: ДМК Пресс, 2004.
- [2] Руководство к Своду знаний по управлению проектами// 3-е изд., PMI, 2004.
- [3] www.pmi.ru.
- [4] www.sovnet.ru.
- [5] Бахиркин М.В. Создание методологии для разработки малых и средних проектов// Материалы VIII Международной конференции по неравновесным процессам в соплах и струях (NPNJ'2010), –М: Изд-во МАИ-ПРИНТ,2010, с.554-556.
- [6] Бахиркин М.В. Создание методологии для разработки малых и средних проектов. // "Новые информационные технологии". Тезисы докладов XVIII Международной студенческой школы-семинара - М.: МИЭМ, 2010, с. 94, 2010 г.
- [7] Бахиркин М.В. Анализ успешности проектов проводимых в государственных научно-исследовательских институтах.//Материалы XVI международной конференции по вычислительной механике и современным прикладным программным системам(ВМСППС'2011), –М: Изд-во МАИ-ПРИНТ,2011, с.180-183.
- [8] Руководство к Своду знаний по управлению проектами.// 4-е изд., PMI, 2004.
- [9] ИСО 10006:2003,Системы менеджмента качества — Руководство по менеджменту качества при проектировании, 2003 год.
- [10] DeMarco T. and Lister T., *Waltzing with Bears* // Dorsett House, 2004 г.
- [11] Руководство к Своду знаний по управлению проектами.// 5-е изд., PMI, 2004.

- [12] Microsoft Solutions Framework, 2002 г.
- [13] Архипенков С.Я. Лекции по управлению программными проектами. //М., 2009 г.
- [14] Бахиркин М.В., Лукин В.Н. Проверим алгеброй гармонию // "Новые информационные технологии". Тезисы докладов XXI Международной студенческой школы-семинара,– М.: МИЭМ НИУ ВШЭ, 2013 г.,– С.45-53.
- [15] Бахиркин М.В. Предпосылки создания методологии разработки IT-проектов в Государственных научно - исследовательских институтах. // Научный Вестник МГТУГА № 184, Москва 2012 г., – С.136-139.
- [16] Little T. Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty// IEEE SOFTWARE, 2006 г. – С. 48 – 54.
- [17] Бахиркин М.В., Зинченко А.С. Квантификация качества IT прогнозов для анализа прогнозируемых данных.// Сборник тезисов и докладов московской молодёжной научно-практической конференция “Инновации в авиации и космонавтике – 2012”, г. Москва,– С 235-236.
- [18] Бахиркин М.В. Квантификация качества IT-прогнозов.// Материалы VIII Международной конференции по неравновесным процессам в соплах и струях (NPNJ'2012), –М: Изд-во МАИ-ПРИНТ,2010 г., – С.589-591.
- [19] Parkinson C.N. Parkinson's Law and Other Studies in Administration //Boston, Houghton-Mifflin, 1957 г, – С 148.
- [20] Frederick P. Brooks The mythical Man-Month: Essays on Software Engineering.// 2010 г.
- [21] David L. Norden-Raleigh Analysis: A Useful Tool for EVM in Development projects. //The Measurable News. 2002 г, – С 24.
- [22] www.ifpug.org
- [23] SO/IEC 20926:2003 Software engineering -- IFPUG 4.1 Unadjusted functional size measurement method -- Counting practices manual , – С 430 p

- [24] Fenton N.E., Pfleeger S.L. Software Metrics: A Rigorous and Practical Approach. // PWS Publishing 2nd edition, revised printing, 1998 г., – С 455 p.
- [25] Колдовский В. Разработка ПО: оценка результата.
- [26] www.ko.com.ua/
- [27] Fazar, W. Program Evaluation and Review Technique. // The American Statistician, Vol. 13, No. 2, 1959 г., – С10.
- [28] C. E. Clark, J. H. Roseboom, Malcolm, D. G., W. Fazar, Application of a Technique for Research and Development Program Evaluation //Operation research, № 5, 1959 г., – С 646-669.
- [29] Kelly J. E., Critical path planning and scheduling: mathematical basis. // Operations research №9, – С 167–179.
- [30] Coates J.F. Technological Forecasting and Social Change (Impact Factor: 1.71). // 08/1999; 62(1). DOI: 10.1016/S0040-1625(99)00013-X, Elsevier Science, 1999 г., – С 235.
- [31] Баценко Д.В. , Василенко Ю.Н.,Сидоров Н.А., Щебетин Ю.В. Модели методы и средства оценки стоимости программного обеспечения. // НАУ. 2004 г., № 7,– С 113 – 118.
- [32] Schofield C. Shepperd M. Estimating software project effort using analogy. // Software Engineering, IEEE Transactions on., Vol 1, 1997 г., – С 736 – 743.
- [33] Kim J. Software cost estimation – Metrics and models.//University of Calgary, Alberta, Canada,. 2001 г. – С115
- [34] Кибзун А.И., Горайнова Е.Р., Наумов А.В. Теория вероятностей и математическая статистика. Базовый курс с примерами и задачами: //Учебное пособие. - 2-е изд., испр. и доп. - М.:ФИЗМАТЛИТ
- [35] Большев Л.Н., Смирнов Н.В. Таблицы математической статистики. //Издательство: Наука, 3-е издание ,1983 г.
- [36] Колмогоров А. Н. Об эмпирическом определении закона распределения. // Т.4, 1933 г.,– С 83-91.

- [37] М.В. Бахиркин., А.С. Зинченко. Квантификация качества IT прогнозов для анализа прогнозируемых данных // Вестник Московского Авиационного Института 2012 г., т.19, №4, – С 181-185.
- [38] CHAOS Report // The Standish Group, 1998 г.
- [39] CHAOS Report // The Standish Group, 2011 г..
- [40] Бахиркин М.В. , Христофоров Г.Ю. Использование технологии SCADE для разработки программного обеспечения бортовой функции поддержки принятия решений. // Девятый Международный симпозиум «Интеллектуальные системы» (INTELS'2010), 2010 г., г. Владимир, – С 252-255.
- [41] Бахиркин М.В, Канадин В.Н., Христофоров Г.Ю. Использование специализированного комплекса SCADE для разработки программного обеспечения бортовой функции поддержки принятия решений// Научный Вестник МГТУГА 2011 г., №184,– С 107-110.
- [42] Боэм Б., Браун Дж., Каспар Х., Липов М., Мак-Леод Г., Мерит М. Характеристики качества программного обеспечения. // М.: Мир, 1981.
- [43] Adamov R. A naive approach to software metrics. // J. Microcomput. AppL, 1989 г., Vol.12, No. 4, – С 343-357.
- [44] DeMarco T. Controlling Software Projects. // Prentice Hall, 1982 г.
- [45] Boehm B.W., Software Engineering Economics. // PrenticeHall, 1981 г.
- [46] Putnam L., Putnam D., Beckert D. A Method for Improving Developers' Software Size Estimates // Crosstalk, 2005 г., vol. 18, no. 4.
- [47] MacCrimmon K.R. and Ryavec C.A. An Analytic Study of the PERT Assumptions //J. Operations Research Soc. of America, vol. 12, no. 1, 1964 г.
- [48] DeMarco T. and Lister T., Waltzing with Bears // Dorsett House, 2004 г.
- [49] Archer C, Stinson M. Object-oriented software measures. //Technical Report. Software Engineering Institute, Carnegie Mellon University. 1995 г.
- [50] Bache, R., Bazzana, G. Software metrics for product assessment. // McGraw-Hill Book Company, 1994 г.

- [51] Briand L.C, Basili V.R., Melo W.L. A validation of object-oriented design metrics as quality indicators // IEEE Trans on Software., 1996 г.
- [52] Belady L.A. Complexity of large systems. Software Metrics: An analysis and evolution // Eds. A.J.Perlis, F.G.Sayward, M.Shaw. Cambridge: M.A.: MIT Press, 1981 г.
- [53] DeMacro T., Lister T. Software development: State of the art vs state of practice. In: IGSF-11, Proc. 11th. Int. Conf. Software Eng., Pittsburg, 1989 г.
- [54] Fenton N.E., Pfleeger S.L. Software metrics :A rigorous and practical approach.
- [55] Second edition. International Tompson Computer Press, 1996 г.
- [56] Harrison W., Magel K., Kluczny R. and DeCock A. Applying software complexity metrics to program maintenance // IEEE Computer. 1982 г. Vol. 15. N9.
- [57] Harrison W. Using software metrics to allocate testing resources // J. Man. Syst, 1988 г. Vol.4: No.4.
- [58] Heitkoetter U., Helling B., Nolte H., Kelly M. Design metrics and aids to their automatic collection // Information and software technology, 1990 г. Vol.32.
- [59] Henderson-Sellers B. Object-oriented metrics : measures of complexity. Prentice Hall PTR, Upper Saddle River, New Jersey, 1996 г.
- [60] Henry S., Kafiira D. Software structure metrics based on infirmation flow // IEEE Trans, on Software Eng. 1981 г., Vol.7, No.5.
- [61] Hirayama M., Sato H., Yamada A., Tsuda J. Practice of quality modeling and measurement on software life-cycle.// In: Proc. 12* Int. Conf Software Eng., Nice, 1990 г., Los Alamitos.
- [62] IEEE standard glossary of software engineering terminology: ANSI/IEEE Std 610.12-1990 г.
- [63] IEEE standard for a software quality metrics methodology: IEEE/ANSI Std 1061,1992 г.

- [64] Kan S. H. Metrics and models in software quality engineering. //Addison-Wesley Publishing Company, 1995г.
- [65] Lorenz M., Kidd J. Object-oriented software metrics : A practical guide. // Prentice Hall, New Jersey, 1994 г.
- [66] McCall J.A., Matsumoto M.T. Software quality metrics enhancements. // Vol.1: RAD-TR-80-109VOL-1. Sunnyvale, Calif: General Electric Co, 1980 г.
- [67] Navlakha J.K. A survey of system complexity metrics // Computer J. Vol. 30, No.3, 1987 г.
- [68] Shepperd M. An evaluation of software product metrics // Inform, and Software Technol., 1988 г., Vol.30, No.3.
- [69] Shepperd M., Ince D. Metrics , outlier analysis and the software design process // Inf and Software Technol., 1989 г., Vol.31, No. 10.
- [70] Shepperd M. Early life-cycle metrics and software quality models // Inf and Software Technol., 1990 г., Vol.32, No.4.
- [71] Shepperd M. Foundation of software measurement // Prentice Hall International, 1995 г.
- [72] Sherif Y.S. Computer software development: Quality attributes, measurements, and metrics // Naval Research Logistics, 1988 г., Vol. 35.
- [73] Fenton N., Whitty R., Iizuka Y. Software Quality Assurance and Measurement: A Worldwide Perspective // International Thomson Computer Press, London, UK, 1995 г.
- [74] Зуев М. Российскую практику проектного управления - в рамки стандарта // СЮ 5, 2001 г., стр. 62-63.
- [75] Дегтярев О.В., Кан А.В., Орлов В.С. Проблемы моделирования процессов выполнения управляемых потоков воздушного движения. ИММОД-2005, г. Санкт-Петербург.
- [76] Вишнякова Л.В., Чуянов Г.А. Моделирование в поддержку принятия перспективных решений по ОрВД и разработка интегрированной

модульной Авионики с новыми функциональными бортовыми приложениями. // Тезисы доклада 3-ей международной конференции “CNS/ATM Авионика”.

[77] Бахиркин М.В., Орлов В.С. Распределенная модель динамической воздушной обстановки. // ИММОД-2009г. Санкт-Петербург.

[78] www.eurocontrol.int/dossiers/single-european-sky

[79] www.faa.gov/nextgen

[80] studopedia.net

[81] RSDN Magazine 4, 2006г.

[82] Крачтен Ф, Кролл П. Руководство по RUP для практиков. Rational Unified Process - это легко. // М.: КУДИЦ-ОБРАЗ, 2004 г., - 432 с. ISBN 5-9579-0019-2.

[83] Бек. К. Экстремальное программирование // СПб.: Питер, 2002. - 224 с. ISBN 5-94723-032-1;

[84] Ларман К. Применение UML 2.0 и шаблонов проектирования // Вильямс, 2-е издание, 2002 г.

[85] CMMI -DEV, V. 1.3, 2010 г.

[86] www.eclipse.org

[87] www.microsoft.com/rus/msdn/msf

[88] citforum.ru/SE/project/scrum/

[89] <http://www.gost.ru>