

УДК 519.876.2

М. В. Бахиркин, А. С. Зинченко, А. П. Кирпичников,
В. Н. Лукин, Д. П. Ткаченко

**МОДЕЛЬ ДИНАМИЧЕСКОЙ ОЦЕНКИ СТОИМОСТНЫХ, ВРЕМЕННЫХ
И ФУНКЦИОНАЛЬНЫХ ПОКАЗАТЕЛЕЙ ПРОЦЕССА ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ
ПРОГРАММ И ПРОГРАММНЫХ СИСТЕМ**

Ключевые слова: проект, управление проектами, рейтинг успешности проектов, конус неопределенности, фактор приближенной оценки качества, SLIM, COCOMO, COCOMO2, Function Point, PERT.

Проводится сравнительный анализ основных методов оценки программных проектов. Выделяются сильные и слабые стороны существующих методик оценивания. На основе проведенного анализа формализуются требования к модели оценивания. Предлагается алгоритм построения математической модели оценивания.

Keywords: project, project management, chaos report, the cone of uncertainty, Estimation quality factor, SLIM, COCOMO, COCOMO2, Function Point, PERT.

Realize comparative analysis of the main methods of evaluation of software projects. Highlighted strengths and weaknesses of existing assessment methods. On the basis of the analysis we formalize requirements for the assessment model. Propose an algorithm for constructing a mathematical model of assessment.

Введение

В крупных организациях каждый год предлагаются сотни ИТ проектов. Так как ресурсы ограничены, только часть проектов получит одобрение и финансирование. Решения руководства базируются на качественных и количественных оценках (затраты, продолжительности, выгода проектов) какие из проектов являются предпочтительней для организации. Важной частью информации для принимающих решения являются оценки, качество которых имеет большое значение. Рассмотрим рядовой сценарий разработки программ и программных систем (ИТ проект). Его успех определяется сбалансированностью таких факторов, как персонал, методологии, границы проекта, время, критерии качества. Предположим, на старте проекта удалось этого добиться. В ходе выполнения проекта что-то меняется, причём, в худшую сторону. Необходимо изменить указанные факторы, чтобы выправить положение. Количество разработчиков увеличивать нельзя, это, как известно, только ухудшает ситуацию, особенно на последних этапах. Переход на новые методологии предполагает их изучение, что, скорее всего, затормозит работу. Сокращение границ проекта может помочь лишь тогда, когда из него исключаются ещё не разработанные функции. Но к концу проекта все функции находятся в стадии значительной готовности, сэкономить можно лишь на тестировании, что, несомненно, приведет к еще большему снижению качества продукта. Остаётся только два фактора: время и качество. Чтобы завершить проект, надо чем-то пожертвовать. Но именно эти параметры и служат критериями успеха проекта.

Очень часто изначально успешный проект становится неудачным. Бытует мнение, что привлекательные возможности всегда связаны с

риском («кто не рискует, тот не пьёт шампанское»). К сожалению, оно поселилось и на благодатной для него почве программного проектирования. В силу молодости, характерной для нашей профессии, и авантюризма характерного для молодости, компании нередко берутся за проекты, связанные со значительными рисками провала, не учитывая потерь от реализации этих рисков. Более того, руководители умудряются не провести элементарные расчёты для оценки условий выполнимости проекта. И проект автоматически попадает в разряд «безнадёжных».

В результате выбрасываются ресурсы, чтобы выполнить проект вдвое быстрее, чем следует, продукт получается некачественным, заказчик недоволен. Для исправления ситуации обычно тратится ресурсов больше, чем потребовалось бы при нормальной работе. И всё равно продукт останется уродливым, а за командой закрепится дурная слава.

Корень зла кроется в неверной оценке. Разработчики либо использовали неправильную методику, не учли существенные факторы или риски, либо срок сдачи был назначен без всяких расчётов, волевым решением.

**Анализ основных стоимостных, временных
и функциональных методов и метрик
разработки программных проектов**

Для создания собственной методики оценки успешности проектов необходимо провести квантификацию. Квантификация есть сведение качественных характеристик к количественным для следующего этапа — измерения, то есть придания результату численного значения. В настоящее время используют различные метрики для анализа прогнозируемых данных в ИТ [1]. Проведем их анализ.

1. Методы, основанные на принципах

Price-to-win. Суть метода состоит в том, что независимо от предполагаемых реальных затрат на разработку проекта, оценка стоимости ПО корректируется в соответствии с пожеланиями заказчика.

Оценка по Паркинсону [2]. Метод основывается на принципе: «Объем работы возрастает в той мере, в какой это необходимо, чтобы занять время, выделенное на ее выполнение» [3].

2. Линейный подход

При линейном подходе стоимость разработки напрямую зависит от количества строк написанного кода, такой метод оценки стимулирует писать большое количество некачественного кода.

3. Software Life-cycle Model (SLIM)

Модель основывается на утверждении, что затраты на разработку ПО распределяются согласно кривым Нордена-Рэйли, которые являются графиками функции, представляющей распределение рабочей силы по времени [5].

4. Функциональные точки (ФТ) и их разновидности

Методика анализа ФТ. Основывается на концепции разграничения взаимодействия. Сущность её состоит в том, что программа разделяется на классы компонентов по формату и типу логических операций [6]. В основе этого деления лежит предположение, что область взаимодействия программы разделяется на внутреннюю – взаимодействие самих компонентов приложения, и внешнюю – взаимодействие с другими приложениями.

Точки свойств. В условиях, когда сформулированные требования не отражают истинной сложности реализации (что характерно для системного ПО, критически важных программных комплексов и пр.), метод функциональных точек себя не оправдывает. В этом случае используют модифицированный вариант – метод точек свойств, который предусматривает корректирование получаемой оценки с учетом алгоритмической сложности.

Метод Mark II. Модификация метода функциональных точек позволяет добиться одного и того же результата как при оценке системы в целом, так и при суммировании оценок, полученных для составляющих ее подсистем.

Трехмерные функциональные точки. Логическое развитие оригинального подхода было предложено подразделением корпорации Boeing. В основу этого метода положена идея о том, что сложность задачи в программной среде можно представить в трех измерениях – данные (количество вводов/выводов), функции (сложность вычислений) и контроль (управляющая логика).

Объектные точки. Поскольку классическая интерпретация метода функциональных точек не предусматривает применения объектно-ориентированного подхода, в современных проектах используется его адаптированный вариант, оперирующий именно терминами ОО-технологии. Существуют приложения, в оценке которых использование стандартных функциональных точек не эффек-

тивно. Эти приложения следующие: управление процессом в реальном времени; математические вычисления; симуляция; системные приложения; инженерные приложения; встроенные системы. Перечисленные приложения отличаются высокой интенсивностью вычислений, часто основанных на алгоритмах повышенной сложности.

5. Program Evaluation And Review Technique (PERT)

Метод PERT использует 3 оценки расчета времени для каждой операции: оптимистическое (наилучшее); средний показатель (наиболее вероятное); пессимистическое (наихудшее). Разработчики PERT для выражения продолжительности операции решили избрать аппроксимацию β -распределения. Распределение проекта представляет собой сумму средневзвешенных показателей операций на критическом пути. Зная среднюю продолжительность проекта и дисперсии (среднего отклонения) операций, можно с помощью статистических таблиц рассчитать выполнение проекта (или сегмента проекта) к конкретному времени.

6. Метод Дельфи (Wideband Delphi)

Основа метода состоит в том, чтобы с помощью серии последовательных действий: опросов; интервью; мозговых штурмов – добиться максимального консенсуса при определении правильного решения. Анализ с помощью дельфийского метода проводится в несколько этапов, результаты обрабатываются статистическими методами.

7. Экспертная оценка

Метод основывается на принципе экспертной оценки и применяется в проектах, использующих новые технологии, новые процессы или решающих инновационные задачи. К процессу оценки привлекаются инженеры-разработчики, которые сами оценивают курируемую ими часть проекта[9].

8. Оценка по аналогии

Проекты и их соответствующие характеристики отображаются в n-мерном пространстве как точки, после чего вычисляется евклидово расстояние между соответствующими точками [10]. Проекты, имеющие наибольшее сходство, будут расположены ближе всего, т. е. евклидово расстояние у них будет наименьшим.

9. COSOMO, COSOMO II, Методика Госкомтруда (86)

В модели трудозатраты нелинейно зависят от размера проекта и скачкообразно изменяются при смене режима. Размер проекта измеряется числом строк кода, функциональными и объектными точками. Помимо прочего, при расчете показателей COSOMO II учитывается уровень зрелости процесса разработки.

10. Конус неопределенности

Конус неопределенности (рис. 1) описывает изменение количества неопределенности в течение проекта. Горизонтальная ось содержит основные вехи проекта: 1 – начальная концепция; 2 – подтверждение концепции продукта; 3 – требования

завершены; 4 – дизайн пользовательского интерфейса завершен; 5 – детальный дизайн закончен; 6 – программное обеспечение готово. Вертикальная ось содержит степень ошибок, которые могут быть совершены экспертом на различных этапах проекта. Эти оценки могут относиться ко всему содержанию проекта – размер усилий, цена, свойства или их комбинация.

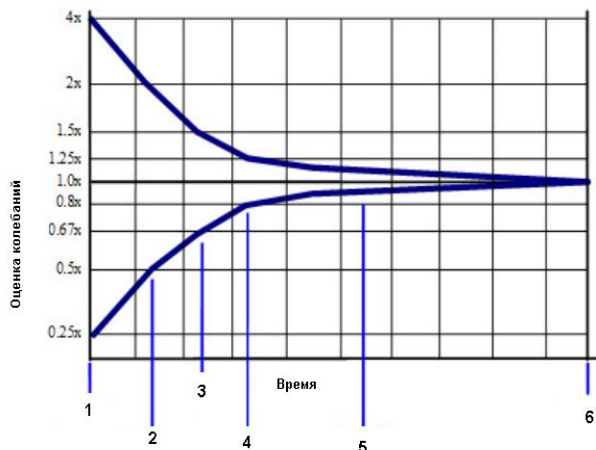


Рис. 1 – Конус неопределенности

11. Метод Де Марко

Метод основан на использовании «бэнг-метрики», близкой по своему содержанию к функциональным точкам. Оценки корректируются с учетом хронологических данных по выполненным ранее проектам, что позволяет получить не абстрактные показатели, а приближенные значения реальных затрат ресурсов и времени.

12. Estimating Quality Factor (EQF)

Фактор приблизительной оценки качества (EQF) является мерой отклонения прогнозируемого значения от реального. Он позволяет измерять качество предварительной оценки. Низкое значение EQF соответствует тому, что все прогнозы были низкого качества и отклонение первоначальных прогнозов огромно. Высокое значение EQF соответствует тому, что все прогнозы были высокого качества и отклонение первоначальных прогнозов минимально.

Резюмируем достоинства и недостатки существующих метрик

Модели, основанные на прямом или косвенном подсчете тысяч строк исходного кода – линейная модель, SLIM, COCOMO, COCOMOII, методика Госкомтруда (86). К достоинствам можно отнести простоту методов; легкость применения; широкую распространённость и известность моделей. К недостаткам – стоимость разработки напрямую зависит от количества строк написанного кода. Стимулирует написание большого количества некачественного кода. Нелинейная зависимость трудоемкости от объема кода. В части моделей сложность предложенных показателей для руководства и финансовых служб. Для COCOMO II в большинстве случаев невозможно получить необходимые оценки с приемлемым уровнем достоверности.

Функциональные точки и их разновидности,

Метод Де Марко. К достоинствам можно отнести тот факт, что измерения не зависят от применяемой технологической платформы, метод обеспечивает единообразный подход к оценке всех проектов в компании, оценка трудозатрат может быть выполнена на самых ранних стадиях работы над проектом. К недостаткам – высокая сложность и трудоемкость метода, большая потеря времени на повторный анализ, который необходимо проводить по мере продвижения разработки.

Экспертные оценки метод PERT. К достоинствам можно отнести простоту и прозрачность метода; К недостаткам – в новых инновационных проектах экспертные оценки случайны из-за отсутствия личного опыта эксперта.

Экспертные оценки Метод Дельфи (Wideband Delphi). Метод дает хорошие результаты при использовании эмпирических данных с адаптацией их в параметрических моделях. Недостатки: уязвимость эксперта перед организационной группой; мнение большинства экспертов может быть принято вместо наиболее эффективного решения; анализ занимает существенное время, что не подходит для оперативного планирования; способствует возрастанию конформизма экспертов; возможность манипуляции экспертами организационной группой; опрашиваемые должны уметь хорошо излагать свои мысли в письменной форме; анкетизируемые должны обладать высоким уровнем мотивации.

Конус неопределенности и EQF. К достоинствам можно отнести простоту и прозрачность метода. Метод позволяет понять тенденцию оценок каждого эксперта. Основные недостатки: обманчивые и односторонние определения; искажение практической оценки; бессмысленные данные; коническая форма конуса не является основанием улучшения оценки.

К суммарным недостаткам всех моделей можно отнести:

- устаревание метрик (разработки 70-90-х годов прошлого века);
- невозможность получить необходимые оценки с приемлемым уровнем достоверности;
- большие трудозатраты;
- большие временные потери на оценку;
- большая потеря времени на повторный анализ;
- отсутствие современных отечественных метрик и методик оценки;
- невозможность (из-за трудоемкости) использование части метрик в проектной работе;
- сложность предложенных показателей для руководства и финансовых служб;
- невозможность использовать существующие метрики и методы в новых инновационных проектах;
- использование экспертных оценок затруднено в связи с высоким конформизмом экспертов и отсутствием независимых аналитических структур и традиции проведения таких исследований.

Оценка трудоемкости должна быть вероятностным утверждением. Это означает, что для нее существует некоторое распределение вероятности,

которое может быть очень широким, если неопределенность высокая, или достаточно узким, если неопределенность низкая. Использование собственного опыта или опыта коллег, полученного в похожих проектах, это наиболее прагматичный подход, который позволяет получить достаточно реалистичные оценки трудоемкости и срока реализации программного проекта, быстро и без больших затрат. Если собственный опыт аналогичных проектов отсутствует, а коллеги-эксперты недоступны, то необходимо использовать формальные методики, основанные на обобщенном отраслевом опыте. Нереалистичность оценок один из серьезнейших демотивирующих факторов для участников проектной команды. Недооценка приводит к ошибкам планирования и неэффективному взаимодействию. Агрессивные сроки, постоянное давление, сверхурочные, авралы служат причиной того, что затраты на проект растут экспоненциально и неограниченно.

Вывод: Необходимо построить базовую модель оценки, подходящую к современным реалиям.

Модель динамической оценки стоимостных, временных и функциональных показателей процесса проектирования и разработки программ и программных систем

Модель динамической оценки стоимостных, временных и функциональных показателей процесса проектирования и разработки программ и программных систем должна:

- давать корректные оценки;
- требовать минимальное количество экспертов для оценки (минимум 3 эксперта);
- быть применимой на практике;
- быть простой и прозрачной для руководства и людей, принимающих финансовые решения;
- не зависеть от объема программных проектов и выбранных методологий управления проектами;
- позволять достичь скорости и точности оценки, достаточной для оперативного управления;
- реализация математической модели должна быть широкодоступна и не требовательна к установке и использованию;
- опираться на лучший мировой опыт; учитывать достоинства и недостатки существующих моделей;
- иметь возможность гибкого учета экспертных оценок;
- должна учитывать вероятностный характер оценок экспертов.

Исходя из анализа, проведенного выше, за базис для создания модели оценивания возьмём:

- модель PERT для первого шага модели;
- принцип Wideband Delphi для построения;
- конус неопределенности и EQF фактор для анализа распределения оценок эксперта.

В рамках своей проектной работы проанализированы данные по трем проектам. На каждом этапе проекта предлагалось оценить итоговый срок разработки проекта в днях. Были выбраны три эксперта, входящие в каждый проект (см. таблицу 1).

Таблица 1 – Характеристики проектов

№	Продолжительность, (мес.)	Количество оценок за весь проект	Количество экспертов всего
1	9	98	4
2	7	102	3
3	20	101	8

Построим конус неопределенности для каждого проекта и эксперта. Для этого по таблице 1 замеров построим график зависимости $K=F/A$ от процента продвижения проекта (см. рис. 2).

Проведем исследование эмпирического распределения $F'(K_T)$ оценок экспертов, используя критерий согласия Пирсона и критерий согласия Колмогорова-Смирнова.

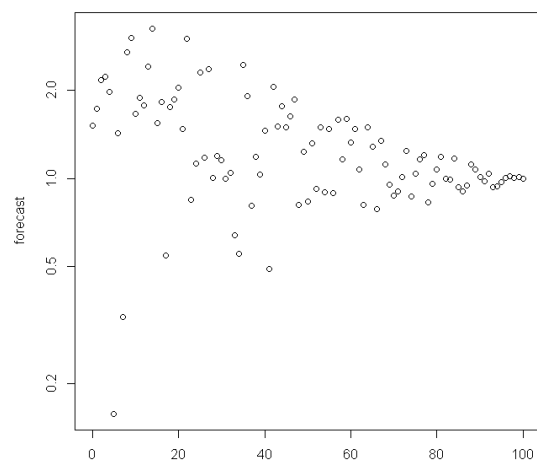


Рис. 2 – Первый проект, K_T эксперта А

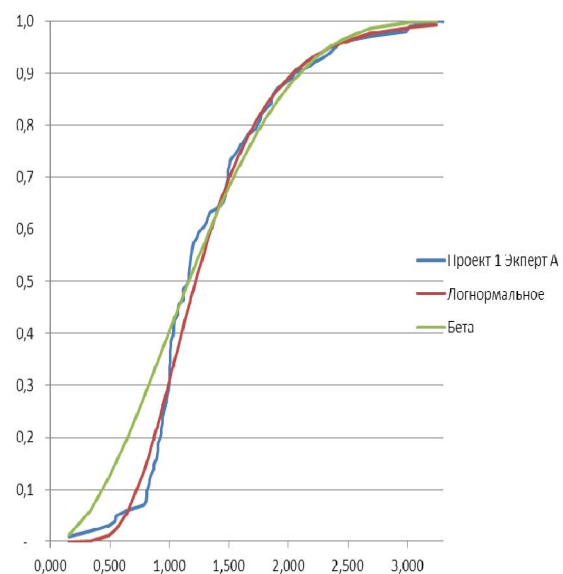


Рис. 3 – Функция распределения: Проект 1, Эксперт А

В результате исследования получаем, что эмпирическое распределение $F'(K_T)$ оценок экспертов по проектам имеет распределение более близкое по своей структуре к логнормальному распределению. Учет этот факт при построении модели оценивания.

Алгоритм построения математической модели оценивания

1. Этап предварительного анализа:

- необходимо разработать иерархическую структуру работ (ИСР);
- определить взаимосвязь операций, используя метод диаграмм предшествования;
- построить сетевую диаграмму.

2. Определение количества предполагаемых экспертов. Для каждой задачи, находящейся на первом уровне (пакет работ), необходимо найти и определить количество предполагаемых экспертов. Для задачи, находящейся на первом уровне иерархии, достаточно оценок 3–5 экспертов.

3. Начальный этап оценки проекта. Для первого этапа анализа необходимо провести оценку проекта по методу PERT.

3.1. Для каждого эксперта k и пакета работ i проекта определяем:

Таблица 2 – Коэффициенты оценки

$A_{i,k}$	Оптимистическое время пакета работ i
$B_{i,k}$	Пессимистическое время пакета работ i
$M_{i,k}$	Наиболее вероятное время пакета работ i

3.2. Для каждого пакета работ внутри фазы считаем средневзвешенное время $t_{i,k}$ по формуле:

$$t_{i,k} = \frac{A_{i,k} + B_{i,k} + 4 * M_{i,k}}{6},$$

где $t_{i,k}$ – мнение k эксперта, i – номер пакета работ.

3.3. Введем коэффициент доверия эксперту k в пакете работ i .

Коэффициент доверия $\alpha_{k,i}$, где для фазы i :

$$\sum_{k=1}^n \alpha_{k,i} = 1, i = \text{const}.$$

На первом этапе можно считать равновероятное мнение всех экспертов. Если количество экспертов в пакете $i=n$, то $\alpha_{k,i}=1/n$. Возможно расставить коэффициенты, согласно предполагаемой компетенции эксперта.

3.4. Введем линейную свертку мнения экспертов $Z_i = \sum_{k=1}^n \alpha_{k,i} \times t_{i,k}, i = \text{const}$ для фазы i .

3.5. Накладываем среднее значение на сеть проекта, рассчитываем раннее, позднее, резервное время завершения проектных работ.

3.6. Рассчитываем: критический путь, плавающее или резервное время, критическую деятельность, время ввода, время запаздывания, резерв времени.

Шаги 3.1–3.6 – соответствуют оценке проекта методом PERT, за исключением оценок расчета временной продолжительности. Основное изменение – ввод линейной свертки. Таким образом, мы учитываем не только мнение руководителя проекта (или единственного эксперта), а суммарное мнение экспертов.

4. Старт проекта.

4.1. Для оценки детализируем актуальную фазу работ на элементарные пакеты работ.

4.2. Для пакета работ i проекта производим оценки длительности для каждого эксперта.

4.3. Рассчитываем коэффициент $K=F/A$ (Forecast/Actual).

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

4.5. Для каждой оценки пакета фазы i , рассчитываем EQF.

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

$$\alpha_{k,i} = \frac{EQF_{k,i}}{\sum_{k=1}^n EQF_{k,i}}.$$

4.7. На основе полученных коэффициентов доверия экспертов $\alpha_{k,i}$, считаем продолжительность остальных пакетов работ фазы i как линейную свертку мнения экспертов:

$$Z_i = \sum_{k=1}^n \alpha_{k,i} \times t_{i,k}, i = \text{const}.$$

4.8. Рассчитываем среднюю продолжительность проекта, имея реальные и прогнозные значения фазы i .

4.9. Накопления полученной информации в базу экспертных оценок.

Для каждой фазы i и пакетов работ внутри фазы накапливаем параметры для базы знаний:

- тип фазы;
- номер пакета работ;
- значение коэффициентов доверия экспертов $\alpha_{k,i}$;
- значение $EQF_{k,i}$.

5. Для фазы 2 повторяем пункты 4.2–4.9.

Заключение

1. При использовании метода PERT средняя продолжительность проекта – это сумма всех средних показателей времени, отведенных на выполнение операций по критическому пути. Все операции, в том числе стоящие на критическом пути, согласно разработчикам метода PERT имеют бета-распределение. Оценки экспертов, проводимые в течение проекта, имеют различный вес, который необходимо учитывать и имеют распределение более близкое по своей структуре к логнормальному распределению.

2. Построенная модель оценивания учитывает тот факт, что оценки экспертов имеют логнормальное распределение. Для текущего момента времени руководитель проекта имеет более точную оценку работ, предстоящих в ближайшей фазе. Это достигается за счет использования экспертных оценок и перестроения весов в зависимости от качества прогноза эксперта.

3. Точность оценки. При использовании предложенной модели оценивания удалось достичь повышения точности оценки на 15–20 %.

Литература

1. М.В. Бахиркин, *Научный вестник МГТУ ГА*, **184**, 136–139 (2011).
 2. B.W. Boehm, *Software engineering economics*, Prentice-Hall, P. 320 (1981).
 3. S.N Parkinson, *Parkinson's Law and Other Studies in Administration*, Houghton-Mifflin, P. 148 (1957).
 4. F. P. Brooks, *The mythical Man-Month: Essays on Software Engineering*, Prentice-Hall, P. 312 (2010).
 5. L. David, *The Measurable News*. P. 24 (2002).
 6. Software engineering: IFPUG 4.1 Unadjusted functional size measurement method: Counting practices manual, ISO/IEC, P. 430 (2003).
 7. N.E. Fenton, S.L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*, PWS Publishing Company, P. 455 (1997).
 8. J. Coates, *Technological Forecasting and Social Change*, Elsevier Science Inc, P. 235 (1999).
 9. M. Shepperd, C. Schofield, Estimating software project effort using analogy, *IEEE Trans Software Eng*, 736–743 (1997).
 10. K. Johnson, Software cost estimation, Metrics and models, University of Calgary, P. 115 (2001).
-
- © **М. В. Бахиркин** – асп. МАИ (НИУ), bakhirkin@mail.ru; **А. С. Зинченко** – к. э. н., зам. проректора по внеучеб. и восп. работе МАИ (НИУ), zinchenko1980@yandex.ru; **А. П. Кирпичников** – д. ф.-м. н., зав. каф. интеллектуальных систем и управления информационными ресурсами КНИТУ, kirpichnikov@kstu.ru; **В. Н. Лукин** – к. ф.-м. н., доцент МАИ (НИУ), lukinvn@list.ru; **Д. П. Ткаченко** – к. т. н., нач. цикла воен. инст. МАИ (НИУ), tdp@list.ru.