



УДК 004.94

Научная статья

<https://doi.org/10.32603/2071-8985-2025-18-10-45-56>

### Экономические аспекты построения киберфизических систем, использующих системы цифровых двойников

**В. Я. Ананьева**

Санкт-Петербургский государственный электротехнический университет

«ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

varvara.spb99@mail.ru

**Аннотация.** Предметом рассмотрения в настоящей статье служат принципы и метрики оценки экономической и качественной составляющих киберфизических систем (КФС), основанных на цифровых нитях (ЦН). Рассматриваются существующие методы оценки стоимости создаваемого программного обеспечения, дается сравнительная оценка данных методов (преимущества/недостатки/применение, а также в зависимости от характеристик системы). Приводятся потенциальные выигрыши и дополнительные издержки, связанные с применением в КФС ЦН, созданных на базе цифровых двойников времени выполнения. Предлагается экономическая модель такой системы. Рассматриваются показатели качества КФС, основанных на ЦН: формулируется общий подход к построению систем метрик и рассматриваются показатели качества КФС, наблюдаемой системы (НС), которая входит в состав КФС, и ЦН, служащей инструментом для взаимодействия заинтересованной стороны с НС. Построена схема классификации метрик ЦН, которые можно классифицировать по группам: уровень применения, уровень детализации объекта оценки, уровень JDL-модели, аспекты, назначение.

**Ключевые слова:** кибер-физическая система, цифровая нить, цифровой двойник, оценка стоимости ПО, методы оценки стоимости ПО, экономическая модель КФС, показатели качества КФС, метрики качества КФС, показатели качества ЦН, метрики качества ЦН, расширенная JDL-модель

**Для цитирования:** Ананьева В. Я. Экономические аспекты построения киберфизических систем, использующих системы цифровых двойников // Изв. СПбГЭТУ «ЛЭТИ». 2025. Т. 18, № 10. С. 45–56. doi: 10.32603/2071-8985-2025-18-10-45-56.

Original article

### Economic Aspects of Building Cyber-Physical Systems Using Digital Twin Systems

**V. Ya. Ananueva**

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

varvara.spb99@mail.ru

**Abstract.** The subject of consideration in this article is the principles and metrics for evaluating the economic and qualitative components of cyber-physical systems (CPS) based on digital threads (DTh). Existing methods of estimating cost of the software are considered, and comparative evaluation of these methods is given (advantages/disadvantages/application, as well as depending on the characteristics of the system). The potential gains and additional costs associated with the use of DTh, created on the basis of run-time digital twins, in CPS, are given. Economic model of such a system is proposed. Quality indicators of CPS based on DTh are consid-

ered: a general approach to the construction of metric systems is formulated and the quality indicators of the CPS, the observed system (OS), which is part of the CPS, and DTh, which is a tool for the interaction of stakeholders with OS, are considered. A classification scheme for the DTh metrics is constructed and can be classified into groups: application level, evaluation object level of detail, JDL-model level, aspects and usage.

**Keywords:** cyber-physical system, digital thread, digital twin, software cost estimation, software cost estimation methods, CPS economic model, CPS quality indicators, CPS quality metrics, digital thread quality indicators, digital thread quality metrics, extended JDL-model

**For citation:** Ananeva V. Ya. Economic Aspects of Building Cyber-Physical Systems Using Digital Twin Systems // LETI Transactions on Electrical Engineering & Computer Science. 2025. Vol. 18, no. 10. P. 45–56. doi: 10.32603/2071-8985-2025-18-10-45-56.

При проектировании и разработке систем необходимо не только иметь в виду техническую сторону вопроса, но также и продумывать экономический аспект. Следует учитывать интересы всех заинтересованных сторон, в данном случае их представляют менеджеры и архитекторы. Архитектору важно, чтобы система удовлетворяла поставленным требованиям, при этом стоимость модулей аппаратной части, дополнительное программное обеспечение не всегда учитываются в архитектурном решении. Архитектурные решения могут влиять на стоимость проекта различными способами:

– архитектору необходимо заложить в систему свойства, за которые готов заплатить покупатель системы;

– архитектор может не ограничивать команду при выборе тех решений, стоимость реализации и поддержания которых невелика.

**Известные подходы.** При разработке КФС различного класса и уровня сложности важность стоимостного фактора может существенно различаться. Например, для систем нижнего ценового спектра, реализуемых большим тиражом, экономический фактор может выходить на первый план. Крупные предприятия, в свою очередь, могут разрабатывать системы высокого уровня сложности, вложения в которые прогнозируются с учетом их использования в долгосрочном периоде, что также требует оценки стоимости продукта с учетом различных факторов.

Методы оценки стоимости создаваемого программного обеспечения могут быть классифицированы на следующие типы:

- экспертные методы [1]–[4];
- методы, основанные на изучении аналогов [1]–[3];
- методы параметрической оценки [1]–[6];
- методы вероятностной оценки [5], [7];

– методы, основанные на использовании алгоритмов машинного обучения [1]–[3], [6].

Экспертные методы можно применять, используя опыт одного или нескольких экспертов (групповая оценка экспертов [7], [8]).

Изучая аналогичные системы, можно также дать оценку стоимости разработки, используя информацию как по своим проектам, так и по проектам других компаний. Чем ближе к аналогу будет создаваемая система, тем оценка будет ближе к реальной стоимости разработки. Однако новая система будет отличаться от аналогов, поэтому, чтобы дать оценку, необходима детальная информация о структуре аналогов, о методологиях их разработки, команде разработки и т. д.

В методах параметрической оценки используются количественные параметры (количество строк кода, функциональные точки и др.). Примерами таких методов могут служить:

– COCOMO (Constructive Cost Model), COCOMO II, основанные на программном обеспечении, измеряемом в тысячах строк кода, учитывают множество таких факторов, как квалификация персонала, сложность реализации, требования к надежности продукта, перспектива повторного использования и др.;

– метод функциональных точек (Function Point Analysis, FPA), предназначенный для оценки объема программного продукта на основе логической модели, учитывающей количество и сложность требуемых функциональных возможностей.

В данных методах учитывается экспертное мнение, а также данные статистики: в методе функциональных точек необходимо понимать сколько необходимо трудозатрат на реализацию одной функциональной точки, если такая информация не ведется, то лучше использовать метод COCOMO II, который позволяет оценить трудоемкость и длительность разработки системы.

К методам оценки суммарной трудоемкости проекта, значение которой не будет превышено с определенной долей вероятности, можно отнести метод PERT (Program/Project Evaluation and Review Technique). Необходимо определить список пакетов работ и для каждого элементарного блока определить: наиболее вероятную оценку трудозатрат; минимально возможную оценку, когда ни один из рисков не осуществился; и пессимистическую оценку трудозатрат, когда все прогнозируемые и непрогнозируемые риски были реализованы. В результате использования данного метода будет получена оценка суммарной трудоемкости проекта, которая не будет превышена с заданной вероятностью, например 95 %.

Для работы с методами, основанными на использовании алгоритмов машинного обучения, необходимо большое количество данных по проектам. При этом должен быть единый стиль получения значений по одним и тем же параметрам – одно и то же значение для различных проектов должно считаться одинаково. При использовании алгоритмов машинного обучения могут быть выявлены скрытые закономерности, которые не были выявлены экспертами, но влияют на сроки и стоимость разработки программных систем.

В каждом случае используются свои методы, поэтому на практике возникает еще один тип – гибридный, когда для оценки стоимости разработки системы применяется несколько методов, значения всех оценок анализируются и появляется обобщенная оценка, которая, например, может быть одновременно основана на экспертных и параметрических оценках.

В табл. 1 приведена сравнительная оценка стоимости создаваемого программного обеспечения.

Выбор метода оценки стоимости разработки программного обеспечения зависит от множества факторов:

- размер и сложность системы;
- степень определенности требований;
- вероятность изменения требований;
- используемые технологии.

Сравнение методов приведено в табл. 2.

Наиболее близкими к КФС, основанным на цифровых нитях, которые позволяют работать с множеством цифровых двойников (ЦД) времени выполнения, рассматриваемых на различных этапах жизненного цикла (ЖЦ) системы, являются линейки или семейства программных продуктов (ЛП). Экономические модели ЛП подробно представлены и проанализированы в [9]. Наиболее популярный показатель эффективности ЛП – ROI

*Табл. 1. Сравнение методов оценки стоимости создаваемого программного обеспечения  
Tab. 1. Comparison of software cost estimation methods*

Метод	Описание	Преимущества	Недостатки	Применение
Экспертные методы	Оценка проводится квалифицированными специалистами на основе опыта и аналогий	Быстрота, гибкость, учет особенностей проекта	Субъективность, зависимость от выбранных экспертов и их числа	Быстрая оценка при нехватке данных; MVP (Minimum Viable Product, минимально жизнеспособный продукт), стартапы
Методы, основанные на изучении аналогов	Использование данных по завершенным схожим проектам	Простота	Зависимость оценки от схожести с аналогичными проектами	Наличие завершенных схожих проектов
Методы параметрической оценки	Работа с моделями, учитывающими значения различных параметров	Гибкость, независимость от технологий	Сложность моделей, требуют учета множества параметров	Крупные проекты, проекты с четко определенной функциональностью; MVP, стартапы
Методы вероятностной оценки	Учет различных типов оценок: оптимистичной, пессимистичной и наиболее вероятной	Уменьшение неопределенности, учет рисков	Необходимость нескольких оценок	Проекты с рисками и неопределенностью
Методы на основе алгоритмов машинного обучения	Прогнозирование на основе исторических данных и алгоритмов	Возможность автоматизации, обнаружение скрытых закономерностей	Необходимость больших данных, сложность интерпретации	Повторяющиеся проекты; MVP, стартапы, если известна статистика

*Табл. 2. Сравнение методов оценки стоимости создаваемого программного обеспечения*

*в зависимости от характеристик системы*

*Tab. 2. Comparison of software cost estimation methods depending on the system characteristics*

Метод	Размер/сложность системы	Степень детализации требований	Вероятность изменения требований	Уровень развития технических решений
Экспертные методы	Маленькие/средние	Низкая	Высокая	Не важен
Методы, основанные на изучении аналогов	Маленькие/средние	Низкая	Средняя/высокая	Есть аналоги
Методы параметрической оценки	Любые	Средняя/высокая	Средняя	Известные или частично известные технологии
Методы вероятностной оценки	Средние/большие	Средняя	Высокая	Может быть неясным
Методы, основанные на использовании алгоритмов машинного обучения	Любые	Средняя/высокая	Может быть низкой	Известные технологии

(Return On Investment, рентабельность инвестиций), который определяется как отношение между экономией (выгодой) и объемом инвестиций. Однако данная оценка не учитывает ставку дисконтирования, поэтому наряду с ROI для оценки эффективности ЛП используют NPV (Net Present Value, чистая приведенная стоимость) – сумму всех дисконтированных денежных потоков от проекта за весь срок его реализации, уменьшенную на сумму первоначальных инвестиций. Т. е. NPV используется для оценки абсолютной прибыли проекта с учетом времени.

**Экономические эффекты от внедрения в КФС ЦН на основе ЦД времени выполнения.** В качестве основных преимуществ использования в КФС ЦН на базе ЦД можно выделить:

1) создание КФС нового уровня сложности (построение более сложных и масштабируемых решений);

2) повышение эффективности функционирования за счет контекстной и контентной адаптации (система становится более гибкой, способной подстраиваться под меняющиеся внешние условия и потребности пользователей);

3) продление срока службы системы (благодаря модульности, динамической обновляемости и возможностям предиктивного обслуживания КФС может эксплуатироваться дольше без полной замены);

4) возможность эффективно строить линейки продуктов на единой платформе (единая цифровая основа позволяет создавать семейство решений, ориентированных на разные задачи, рынки или уровни сложности);

5) уменьшение затрат на поддержание НС (за счет удаленного мониторинга, диагностики и мо-

делирования можно заранее выявлять и устранять потенциальные проблемы, снижая стоимость обслуживания);

6) уменьшение затрат на модернизацию системы (гибкость, заложенная в архитектуре КФС, позволяет удовлетворить новые потребности, а также снизить стоимость модернизации);

7) возможность повысить эффективность производств за счет использования ЦД (использование ЦД времени выполнения позволяет проводить моделирование «что-если», оптимизировать процессы в режиме реального времени и повышать эффективность управления);

8) повышение уровня автоматизации (за счет использования моделей и динамической адаптации можно реализовать более высокий уровень автономного управления, таким образом уменьшается необходимость вмешательства человека в автоматизируемые процессы).

К основным дополнительным издержкам можно отнести следующие:

1) концептуальная сложность, вследствие чего высокие начальные затраты на внедрение (разработка моделей, создание платформы, интеграция с существующими системами требуют значительных инвестиций);

2) увеличение сроков разработки (время вывода первой версии увеличивается, однако последующие версии создаются быстрее);

3) необходимость разработки и поддержания в актуальном состоянии моделей (как входящих в состав системы ЦД времени выполнения, так и другие модели);

4) вероятностный характер получения выигрыша (экономия и эффекты от внедрения зависят от множества факторов);

5) необходимость мощной вычислительной инфраструктуры (реализация механизмов ЦН и ЦД времени выполнения требует увеличенных вычислительных ресурсов, особенно при работе с большими объемами данных и в режиме реального времени);

6) проблемы при работе в режиме реального времени (обеспечение надежного и своевременного взаимодействия между физическими и цифровыми сущностями требует решения ряда технических и архитектурных задач).

**Предлагаемая модель.** Основывается на экономических моделях для ЛП. В данном случае речь идет не о методах оценки стоимости разработки программного продукта, которые ориентированы на количественную оценку трудозатрат и сроков для отдельного проекта, а переходим к экономическим моделям, которые позволяют решить более широкий круг задач – прогноз затрат на весь ЖЦ ЛП, оценка экономики масштабирования, учет коэффициента повторного использования, учет связи качества и стоимости и др.

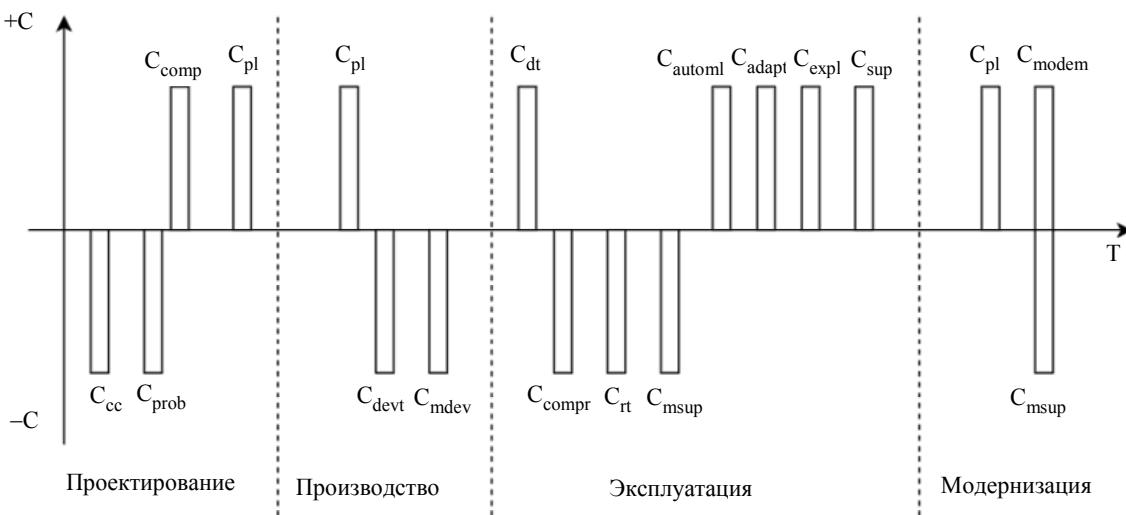


Рис. 1. Динамика денежных потоков на протяжении ЖЦ системы  
Fig. 1. Cash flow dynamics throughout system life cycle

Табл. 3. Потенциальные выигрыши (доходы)

Tab. 3. Potential gains

Обозначение	Источник выигрыша	Способ оценки
$C_{comp}$	Выход на новые рынки за счет возможности создания КФС нового уровня сложности	Сравнение с системами более низкого уровня сложности
$C_{pl}$	Повышение эффективности за счет использования ЛП	Для получения данных оценок можно воспользоваться известными методиками [10]
$C_{adapt}$	Повышение эффективности функционирования за счет контекстной и контентной адаптации	Либо с помощью требуемой функциональности с использованием менее мощных, гибких ресурсов, либо реализацией дополнительной функциональности при одинаковых ресурсах (вычислительной мощности и пропускной способности каналов связи)
$C_{dt}$	Повышение эффективности эксплуатации за счет использования ЦД	Рассмотрение систем, использующих ЦН, которые основаны на трансформации системы ЦД, и не использующих данный инструмент
$C_{automl}$	Уменьшение затрат на персонал за счет повышения уровня автоматизации	Рассмотрение систем, использующих модели и динамическую адаптацию, и не использующих такие механизмы, сравнивая количество операторов и их функции
$C_{expl}$	Продление срока службы системы	С помощью сравнительных оценок стоимости эксплуатации КФС
$C_{sup}$	Уменьшение затрат на поддержание НС	С помощью сравнительных оценок стоимости обслуживания НС с применением систем удаленного мониторинга и моделирования, а также без них
$C_{modern}$	Уменьшение затрат на модернизацию системы	С помощью сравнительных оценок стоимости очередной модернизации КФС, основанных на ЦН (где за счет наличия гибкости можно также с минимальными затратами удовлетворить новые потребности заинтересованных сторон) и не использующих ЦН

Табл. 4. Потенциальные издержки (затраты)  
 Tab. 4. Potential costs

Обозначение	Источник дополнительных затрат	Способ оценки
$C_{cc}$	Концептуальная сложность технических решений	Начальные затраты на внедрение можно оценить за счет дополнительных расходов на разработку и модернизацию ЦН на экономических моделях для линеек продуктов [10]
$C_{devt}$	Увеличение сроков разработки	Сроки разработки можно оценить, например, с помощью методов СОСМО II (параметрическая модель оценки трудозатрат, основанная на размере системы и факторах масштаба/сложности), PERT (основанный на различных уровнях оценок), FPA (анализ функциональных точек, независимый от технологии реализации, используется на этапе анализа требований, когда нет информации о количестве строк кода)
$C_{prob}$	Вероятностный характер получения выигрыша	В некоторые архитектурные состояния КФС может переходить редко, что уменьшает выигрыши и не позволяет окупить затраты на их поддержание
$C_{mdev}$ , $C_{msup}$	Необходимость разработки ( $C_{mdev}$ ) и поддержания в актуальном состоянии ( $C_{msup}$ ) моделей	СОСМО II (оценка трудозатрат на одну модель), SIMPLE (экономика ЖЦ), qCOPLIMO (оценка качества в экономике ЛП), СОСМО PL (оценка стоимости платформы и продуктов на ее основе) – с помощью данных методов и моделей, а также моделирования можно оценить перестроение и создание заново моделей
$C_{compr}$	Необходимость использования более производительных вычислительных механизмов для реализации сложных систем управления	Оценка посредством моделирования необходимой мощности вычислителей и требуемых объемов памяти
$C_{rt}$	Проблемы при работе в режиме реального времени	Оценка посредством моделирования получаемой задержки при обращении не напрямую к НС, а к моделям

Таким образом, предлагаемая модель основывается на экономических моделях для линеек продуктов – таких, как SIMPLE (цикл жизни и финансовые потоки), qCOPLIMO (множество оценок и управление качеством), СОСМО PL (стоимость единой платформы и ее продуктов), которая, в свою очередь, базируется на классической модели СОСМО II [10]. Перечисленные модели адаптированы с учетом специфики ЦН, основанных на системе ЦД.

Экономическая модель КФС, основанной на ЦН, предлагается определить как

$$CM = \langle CFPh, CFT, \{C_i\} \rangle,$$

где

$CFPh$  – денежный поток (Cash Flow), привязанный к этапам ЖЦ (Phases);

$CFT$  – денежный поток, привязанный к временным интервалам (T) этапов  $CFPh$ ;

$\{C_i\}$  – множество оценок, привязанных к временным интервалам  $CFT$ .

На рис. 1 представлена динамика денежных потоков на протяжении ЖЦ системы. На рис. 1 объединены модели  $CFPh$  (денежный поток, привязанный к этапам ЖЦ) и  $CFT$  (денежный поток, привязанный к временным интервалам). По гори-

зонтальной оси отложено время, а по вертикальной – движение денежных средств (+ $C$  – для доходов,  $-C$  – для затрат). Для каждой системы порядок появления доходов и затрат будет свой, на рисунке приведен один из возможных вариантов.

В табл. 3 и 4 приведены расшифровки условных обозначений и возможные способы получения соответствующих оценок.

Список приведенных параметров нельзя назвать полным, но перечисленные параметры практически всегда присутствуют в процессе проектирования реальных систем.

При разработке КФС на основе ЦН начальные затраты будут выше, чем в существующих решениях, долгосрочные выгоды появляются позже, поэтому важно учитывать временную стоимость денег. Для этого можно использовать формулу расчета дисконтированных денежных потоков

$$DCF = \sum_{i=1}^n \frac{CF_i}{(1+r)^i},$$

где  $n$  – количество периодов;  $CF_i$  – денежный поток (приведенная к текущему моменту стоимость будущих поступлений или затрат) в период времени  $i$ ;  $r$  – ставка дисконтирования.

Затем, чтобы понять, рентабелен ли проект, необходимо рассчитать NPV проекта, т. е. чистую приведенную стоимость

$$NPV = DCF - CF_0,$$

где  $DCF$  – сумма дисконтированных денежных потоков за  $n$  периодов;  $CF_0$  – текущие (начальные) инвестиции.

**Показатели качества КФС, основанных на ЦН. Общий подход к построению систем метрик.** Под качеством модели обычно понимают «совокупность свойств и характеристик модели, которые определяют ее способности удовлетворять обусловленные или предполагаемые потребности заинтересованной стороны (субъекта), использующий данную модель» [11], [12]. Т. е. качество системы – степень удовлетворенности заинтересованных сторон.

В зависимости от точки зрения на КФС и ЦН возможны по меньшей мере три взаимодополняющих подхода к построению систем метрик, т. е. показатели качества можно построить:

- 1) для КФС в целом;
- 2) для НС в составе КФС;
- 3) для ЦН, которая служит инструментом для взаимодействия заинтересованной стороны с НС.

Качество КФС в целом можно оценить, используя, например, показатели, описанные в [12]: надежность, простота (сложность), устойчивость, развиваемость, корректность, точность, полезность. КФС также можно оценить и с помощью метрик, используемых для оценки качества НС и ЦН, речь о которых пойдет далее.

Показатели качества НС сильно зависят от природы НС и задач, решаемых с помощью КФС. В общем виде для оценки качества НС можно использовать серию стандартов SQuaRE («Software Product Quality Requirements and Evaluations», «Системная и программная инженерия – Требования к качеству и оценка программного продукта»). В рамках данных стандартов качество системы определяется как «степень удовлетворенности заинтересованных сторон». Для оценки качества систем используются многоуровневые иерархические модели качества, в которых выделяются:

- характеристики (т. е. показатели, обобщающие свойства системы);
- подхарактеристики (метрики, более детализированные аспекты свойств);
- атрибуты (конкретные значения подхарактеристик).

Основой для построения таких моделей служит международный стандарт ISO/IEC 25010:2011 (ГОСТ Р ИСО/МЭК 25010–2015), который определяет два ключевых направления для анализа:

- 1) модель качества в использовании;
- 2) модель качества самой системы.

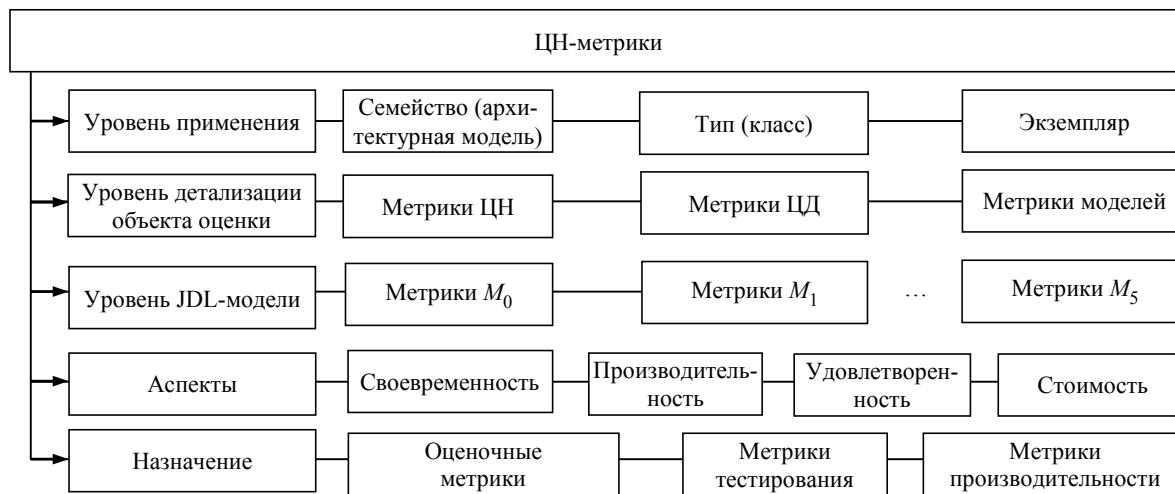
К показателям качества в использовании можно отнести результативность (effectiveness), эффективность (efficiency), удовлетворенность (satisfaction), свободу от риска (freedom from risk), покрытие контекста (context coverage). К показателям качества системы относятся восемь характеристик, каждая из которых обладает своими подхарактеристиками: функциональное соответствие (functional suitability), эффективность функционирования (performance efficiency), совместимость (compatibility), удобство использования (usability), надежность (reliability), защищенность (security), сопровождаемость (maintainability), переносимость (mobility, portability).

**Показатели качества ЦН.** Для оценки качества ЦН можно использовать различные показатели, в том числе серию стандартов SQuaRE, но применительно к ЦН данные показатели можно конкретизировать и ранжировать.

Для показателей (систем метрик) качества ЦН могут быть сформулированы следующие принципы:

- все показатели относятся к определенным архитектурным перспективам, точкам зрения;
- показатель должен быть полезен хотя бы одной заинтересованной стороне, будь то человек, участвующий в управлении, или другая система, использующая данные для принятия решений;
- частота использования метрики, ее актуальность – отдельный уровень анализа, метаметрика;
- показатель выступает как атрибут определенного представления системы, при этом одна архитектурная модель может содержать множество показателей и, следовательно, метрик;
- система метрик должна покрывать ключевые аспекты функционирования системы, но не содержать дублирующих и излишне детализированных показателей;
- способ представления метрик должен соответствовать восприятию заинтересованной стороны;
- затраты на сбор и использование метрики должны быть сопоставимы с ее информационной ценностью [13].

Далее будем рассматривать нижний уровень – метрики. На рис. 2 приведена схема классифика-



*Ruc. 2. Схема классификации метрик ЦН  
 Fig. 2. The classification scheme of digital thread metrics*

ции системы метрик ЦН, которые могут быть разделены на группы: уровень применения, уровень детализации объекта оценки, уровень JDL\*-модели, аспекты, назначение.

По уровню применения метрики можно разделить:

- на метрики семейств (архитектурных моделей);
- метрики типов (или классов);
- метрики экземпляров (конкретных объектов).

В зависимости от объекта оценки метрики могут относиться:

- к ЦН (как системам ЦД, которые относятся к разным этапам ЖЦ и связаны между собой информационными связями);
- ЦД (как элементу ЦН);
- модели (как основе ЦД).

Метрики могут быть привязаны к конкретному уровню JDL-модели (JDL Data Fusion Model – JDL-модель слияния данных). Далее об этом будет подробно рассказано.

Если рассматривать ЦН как систему слияния данных (Data Fusion System, DF) [14]–[15], то можно выделить четыре основные группы параметров (аспектов), которые соответствуют четырем основным группам требований, специфичным для всех DF-систем и ЦН в частности:

- своевременность получения информации (отражает задержку между сбором данных с НС и их обработкой в цифровом представлении и дальнейшей выдачей заинтересованной стороне);
- производительность (эффективность работы системы в целом, в том числе пропускная способность системы (подсистемы));

– удовлетворенность заинтересованных сторон полученной информацией (насколько выводы системы воспринимаются пользователями или техническими компонентами как надежные и полезные);

– стоимость (затраты на создание, внедрение, эксплуатацию и поддержание ЦН, выраженные в денежных единицах или количестве ресурсов).

В данном случае уделяется внимание аспектам, отражающим режим работы в реальном времени и возможность работы множества заинтересованных сторон, т. е. тем характеристикам, которые свойственны DF-системам – системам, которые работают с разнородной информацией, объединенной из разных источников; функционируют в условиях неопределенности и высокой динамики; получают данные в режиме реального времени.

В зависимости от назначения, т. е. того, какая категория специалистов выступает потребителем метрик, метрики можно разделить на группы:

- оценочные метрики (применяются специалистами, занимающимися проектированием и разработкой ЦН: архитекторами, разработчиками);
- метрики тестирования (применяются аналитиками, архитекторами и другими специалистами при сравнении различных реализаций ЦН и выборе наиболее подходящего решения, соответствующего требованиям системы);
- метрики производительности (показывают, насколько НЦ влияет на скорость реакции, качество и уверенность в принятии решений со стороны лица, принимающего решение).

**Метрики уровней JDL-моделей.** Возьмем расширенную структуру, состоящую из шести уровней [14]:

\* Joint Directors of Laboratories – подразделение министерства обороны США, создавшее модель.

$L_0$  (Source Preprocessing) – подготовка данных от датчиков и сенсоров;

$L_1$  (Object Refinement) – объединение данных для идентификации объектов;

$L_2$  (Situation Refinement) – анализ текущих ситуаций и взаимодействий;

$L_3$  (Impact Refinement) – прогнозирование последствий действий и их влияние на систему и «внешний мир»;

$L_4$  (Process Refinement) – управление процессом слияния: оптимизация алгоритмов, адаптация к новым условиям, обратная связь;

$L_5$  (User Refinement) – адаптация системы под пользователя, его интересы.

Далее будут описаны метрики данных уровней применительно к ЦН.

Метрики  $M_0$  – это метрики уровня  $L_0$ , относящиеся к «сырым» логам, которые не прошли очистку. Обычно данные метрики связаны с оценкой полноты и достоверности информации, содержащейся в «сырых» логах. Это могут быть, например, меры степени зашумленности или полноты получаемой информации.

Метрики  $M_1$  ориентированы на оценку способности системы обнаруживать, распознавать и идентифицировать состояние НС или их компонентов, а также другие, внешние по отношению к системе, объекты. Данный уровень связан с анализом объектов, выделением их характеристик и интерпретацией полученных данных. Можно выделить следующие метрики, которые применимы к ЦН:

- вероятность обнаружения (отношение числа обнаруженных состояний/объектов к общему числу состояний/объектов);

- доля успешного распознавания (соотношение между количеством верных распознаваний состояний/объектов и числом попыток);

- точность идентификации (соотношение между количеством верных идентификаций и общим числом попыток).

При этом под обнаружением объекта понимается определение наличия объекта, при распознавании происходит классификация данного объекта, т. е. отнесение его к определенной группе объектов, а при идентификации – установление уникальных характеристик, позволяющих точно определить, какой объект и в каком состоянии находится.

Метрики  $M_2$  обеспечивают интерпретацию информации, исходя из потребностей заинтересованных сторон. Основная цель заключается в

описании текущих или исторических ситуаций и предоставлении информации по ним, а также в предоставлении общей оценки состояния системы или конкретных ее аспектов. На данном уровне происходит переход от анализа отдельных объектов, их состояний к пониманию ситуации в целом: ситуация включает множество объектов, их атрибуты (свойства), отношения между объектами; ситуация может быть абстрактной (общая модель связей и свойств) и реальной (конкретные значения параметров в данный момент времени). Т. е. ситуация – это обобщенное представление, где учитываются не только отдельные состояния объектов, но и их взаимодействия, а состояние объекта – это конкретные данные об одном элементе системы.

Можно выделить следующие метрики, которые применимы к ЦН:

- скорость распознавания ситуации (сколько времени прошло от поступления данных до формирования модели ситуации);

- степень покрытия ситуации (доля параметров системы, которая была учтена при формировании модели ситуации);

- точность анализа ситуации (насколько модель ситуации совпадает с реальностью);

- частота изменений ситуации (сколько раз система за определенное время меняла свое понимание ситуации);

- время жизни ситуации (медианное время актуальности ситуации до ее изменения или завершения).

На уровне  $L_3$  применительно к DF-системам и, в частности, ЦН происходит переход от анализа ситуаций к оценке последствий, которые данные ситуации вызовут или могут вызвать. Т. е. с помощью метрик  $M_3$  происходит оценка анализа событий, которые могут произойти дальше, и их влияния на НС и внешний мир.

Можно выделить следующие метрики применительно к ЦН:

- точность прогноза (отношение верных предсказаний к общему числу событий);

- горизонт прогноза (за сколько времени до появления события система может его спрогнозировать с заданной вероятностью наступления);

- уязвимость (вероятность неблагоприятного события или доля неблагоприятных исходов среди всех возможных, с учетом их вероятностей и весов);

- живучесть (способность системы адаптироваться к изменениям, восстанавливаться после воздействия и продолжать выполнять свои функции).

ции, даже частично, в новых условиях; рассчитывается как доля ситуаций, в которых система сохранила работоспособность);

– временное устаревание ЦН (мера снижения актуальности или точности информации со временем; устаревание модели снижает уровень доверия к выводам, увеличивает неопределенность и может привести к ошибочным действиям).

На уровне  $L_4$  анализируется сам процесс слияния (можно сказать Meta-Process Level [14]), система анализирует и модифицирует свои собственные процессы, т. е. управляет собой. В отличие от метрик предыдущих уровней, где акцент делается на объектах, состояниях, ситуациях и их последствиях, метрики  $M_4$  ориентированы на оценку самонастройки и адаптации алгоритмов, что позволяет системе повышать свою зрелость и устойчивость к изменениям. Таким образом, подсистема управления процессом слияния может быть оценена по следующим ключевым характеристикам, аналогичным тем, которые применяются ко всей системе:

- своевременность (способность системы быстро реагировать на изменение условий и обновлять параметры);
- точность (корректность выбора и применения алгоритмов слияния);
- пропускная способность (объем обрабатываемых данных за единицу времени);
- стоимость реализации (экономические затраты на адаптацию и оптимизацию процессов);
- уровень доверия (степень уверенности в том, что изменения в процессе слияния действительно улучшают работу системы).

Метрики  $M_5$  ориентированы на оценку эффективности системы с точки зрения оператора – лица, принимающего решения. Данный уровень отражает не только техническую сторону ЦН, но и то, как информация воспринимается и используется заинтересованными сторонами, особенно в условиях сложных или критических ситуаций.

На уровне  $L_5$  можно выделить следующие метрики применительно к ЦН:

– степень удовлетворенности пользователя системой (комплексная мера, отражающая восприятие полезности, удобства и надежности ЦН);

– уровень доверия к выводам (характеризует частоту, с которой пользователь принимает информацию, выдаваемую системой, как достоверную);

– скорость принятия решения – время принятия решения пользователем с использованием информации, выданной системой, и без нее;

– количество ошибок оператора – позволяет оценить, насколько использование системы снижает количество ошибок пользователя;

– уровень когнитивной нагрузки – насколько сложно пользователю воспринимать предоставляемую системой информацию.

По рассмотренным метрикам качества можно получить значения и затем учесть их в денежных потоках предложенной экономической модели.

**Заключение.** Таким образом, КФС, основанную на использовании ЦН, можно оценить с различных точек зрения. Для того, чтобы оценить экономическую эффективность будущей системы предложен вариант, основанный на экономических моделях для ЛП. В качестве составляющих данной модели выступают: денежный поток, привязанный к этапам ЖЦ; денежный поток, привязанный к временным интервалам этапов; множество оценок, привязанных к временным интервалам. На каждом этапе ЖЦ системы появляются выигрыши (доходы) и издержки (затраты), которые можно оценить и получить предварительную информацию о том, как будет развиваться система, в какой момент времени необходимо привлечь больше ресурсов, а когда система начнет приносить доход.

Помимо экономических аспектов, перед тем как создавать КФС, необходимо продумать показатели качества, по которым будет оценена степень удовлетворенности каждой заинтересованной стороны. При этом оценить с точки зрения качества можно КФС в целом, НС, которая входит в состав КФС, и ЦН, которая служит инструментом взаимодействия заинтересованной стороны с НС. Наибольший интерес в данной статье представляют метрики качества ЦН, для которых сформулированы принципы построения и использования, а также предложена схема классификации.

Метрики качества служат для учета влияния качества на затраты и выгоды. Таким образом, при оценке экономической составляющей важно учитывать качество системы, уровень которого напрямую влияет на денежные потоки.

#### Список литературы

1. Trendowicz A., Jeffery R. Software project effort estimation: Foundations and best practice guidelines for

success. Switzerland: Springer International Publishing, 2014. 469 p.

2. Briand L. C., Wieczorek I. Resource estimation in software engineering // Encyclopedia of Software Engin. New York: John Wiley & Sons, Inc, 2002. 1929 p.
3. Медведкова И. В., Иванов А. А. Сравнительный анализ методов оценки стоимости проектов по разработке программного обеспечения // Экономические науки. 2019. Вып. 6-1. С. 138–141. doi: 10.24411/2500-1000-2019-11266.
4. Оценка стоимости разработки программного продукта, информационной системы, сервиса или задачи // Хабр. 2023. URL: <https://habr.com/ru/articles/713998/> (дата обращения: 06.06.2025).
5. Basic COCOMO // Cost Estimating Web Site. URL: <https://web.archive.org/web/20111126011102/http://cost.js.c.nasa.gov/COCOMO.html> (дата обращения: 06.06.2025).
6. Оразов Б. Подходы к расчетам стоимости разработки программного обеспечения // Профессорский журнал. Сер.: Экономические науки. 2025. № 1 (5). С. 22–29. doi: 10.18572/3034-2341-2025-1-5-22-29.
7. Бурков Е. А., Падерно П. И. Подход к построению модели объекта экспертизы с возможностью корректировки компетентности экспертов и оценки деятельности рабочей группы. Часть 1. Метод построения атрибутивной модели // Изв. СПбГЭТУ «ЛЭТИ». 2025. Т. 18, № 5. С. 33–44. doi: 10.32603/2071-8985-2025-18-5-33-44.
8. Бурков Е. А., Падерно П. И. Подход к построению модели объекта экспертизы с возможностью корректировки компетентности экспертов и оценки
- деятельности рабочей группы. Часть 2. Методы корректировки компетентности экспертов и оценки деятельности рабочей группы // Изв. СПбГЭТУ «ЛЭТИ». 2025. Т. 18, № 6. С. 47–57. doi: 10.32603/2071-8985-2025-18-6-47-57.
9. Economics-driven software architecture / I. Mistrik, R. Bahsoon, R. Kazman, Yu. Zhang. Burlington, Massachusetts: Elsevier Inc. 2014. 380 p.
10. Software cost estimation with COCOMO II / B. Boehm, C. Abts, A. W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, B. Steele. Englewood Cliffs, NJ: Prentice-Hall, 2000. 544 p.
11. Бутырский Е. Ю. Методы моделирования и оценивания случайных величин и процессов. СПб.: Стратегия будущего, 2020. 641 с.
12. Микони С. В., Соколов Б. В., Юсупов Р. М. Квалиметрия моделей и полимодельных комплексов. М.: Наука, 2018. 312 с.
13. Жукова Н. А. Многоуровневый синтез автоматных моделей объектов мониторинга: дис. ... д-ра техн. наук // СПИИРАН. СПб., 2019. 286 с. URL: [http://www.spiiras.nw.ru/dissovets/wp-content/uploads/2019/11/dis\\_zhukova\\_final03.pdf](http://www.spiiras.nw.ru/dissovets/wp-content/uploads/2019/11/dis_zhukova_final03.pdf) (дата обращения: 06.06.2025).
14. Blasch E., Bosse E., Lambert D. High-level information fusion management and system design. Norwood, MA: Artech House Publ., 2012. 364 p.
15. Liggins M. E., Hall D. V., Llinas J. Handbook of multisensor data fusion: theory and practice. 2<sup>nd</sup> ed. Boca Raton, FL: CRC Press, 2009. 850 p.

## Информация об авторе

**Ананьева Варвара Яновна** – аспирант, ст. преподаватель кафедры информационных систем СПбГЭТУ «ЛЭТИ».  
 E-mail: varvara.spb99@mail.ru  
<https://orcid.org/0009-0000-2074-5438>

## References

1. Trendowicz A., Jeffery R. Software project effort estimation: Foundations and best practice guidelines for success. Switzerland: Springer International Publishing, 2014. 469 p.
2. Briand L. C., Wieczorek I. Resource estimation in software engineering // Encyclopedia of Software Engin. New York: John Wiley & Sons, Inc, 2002. 1929 p.
3. Medvedkova I. V., Ivanov A. A. Sravnitel'nyj analiz metodov ocenki stoimosti proektor po razrabotke programmnogo obespechenija // Jekonomicheskie nauki. 2019. Vyp. 6-1. S. 138–141. doi: 10.24411/2500-1000-2019-11266. (In Russ.).
4. Ocenka stoimosti razrabotki programmnogo produkta, informacionnoj sistemy, servisa ili zadachi // Habr. 2023. URL: <https://habr.com/ru/articles/713998/> (data obrashhenija: 06.06.2025). (In Russ.).
5. Basic COCOMO // Cost Estimating Web Site. URL: <https://web.archive.org/web/20111126011102/http://cost.js.c.nasa.gov/COCOMO.html> (data obrashhenija: 06.06.2025).
6. Orazov B. Podhody k raschetam stoimosti razrabotki programmnogo obespechenija // Professorskij zhurn. Ser.: Jekonomicheskie nauki. 2025. № 1 (5). S. 22–29. doi: 10.18572/3034-2341-2025-1-5-22-29. (In Russ.).
7. Burkov E. A., Paderno P. I. Podhod k postroeniju modeli ob#ekta jekspertizy s vozmozhnostju korrektirovki kompetentnosti jekspertov i ocenki dejatel'nosti rabochej gruppy. Chast' 1. Metod postroenija atributivnoj modeli // Izv. SPbGETU «LETI». 2025. T. 18, no. 5. S. 33–44. doi: 10.32603/2071-8985-2025-18-5-33-44. (In Russ.).
8. Burkov E. A., Paderno P. I. Podhod k postroeniju modeli ob#ekta jekspertizy s vozmozhnostju korrektirovki kompetentnosti jekspertov i ocenki dejatel'nosti rabochej gruppy. Chast' 2. Metody korrektirovki kompetentnosti jekspertov i ocenki dejatel'nosti rabochej gruppy // Izv. SPbGETU «LETI». 2025. T. 18, no. 6. S. 47–57. doi: 10.32603/2071-8985-2025-18-6-47-57. (In Russ.).
9. Economics-driven software architecture / I. Mistrik, R. Bahsoon, R. Kazman, Yu. Zhang. Burlington, Massachusetts: Elsevier Inc. 2014. 380 p.

10. Software cost estimation with COCOMO II / B. Boehm, C. Abts, A. W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, B. Steele. Englewood Cliffs, NJ: Prentice-Hall, 2000. 544 p.
11. Butyrskij E. Ju. Metody modelirovaniya i ocenivaniya sluchajnyh velichin i processov. SPb.: Strategija budushhego, 2020. 641 s. (In Russ.).
12. Mikoni S. V., Sokolov B. V., Jusupov R. M. Kvalimetrija modelej i polimodel'nyh kompleksov. M.: Nauka, 2018. 312 s. (In Russ.).
13. Zhukova N. A. Mnogourovnevyyj sintez avtomatyh modelej ob#ektov monitoringa: dis. ... d-ra tehn. nauk // SPIIRAN. SPb., 2019. 286 s. URL: [http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2019/11/dis\\_zhukova\\_final03.pdf](http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2019/11/dis_zhukova_final03.pdf) (data obrashchenija: 06.06.2025). (In Russ.).
14. Blasch E., Bosse E., Lambert D. High-level information fusion management and system design. Norwood, MA: Artech House Publ., 2012. 364 p.
15. Liggins M. E., Hall D. V., Llinas J. Handbook of multisensor data fusion: theory and practice. 2<sup>nd</sup> ed. Boca Raton, FL: CRC Press, 2009. 850 p.
- 

**Information about the author**

**Varvara Ya. Ananeva** – Postgraduate student, Senior lecturer of the Department of Information Systems of Saint Petersburg Electrotechnical University.

E-mail: varvara.spb99@mail.ru

<https://orcid.org/0009-0000-2074-5438>

Статья поступила в редакцию 22.07.2025; принята к публикации после рецензирования 02.10.2025; опубликована онлайн 30.12.2025.

Submitted 22.07.2025; accepted 02.10.2025; published online 30.12.2025.

---