

---

# СЕКЦІЯ ТЕОРЕТИЧНІ ОСНОВИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

---

UDC 004.415.25

## DESIGNING OF FRAMEWORK FOR UML USE-CASE TO COMMUNICATION DIAGRAM TRANSFORMATION

*Olena Chebanyuk\**, *Juan Bautista Castellanos Peñuela\*\**

\* chebanyuk.elena@ithea.org, National Aviation University, Ukraine

\*\* jcastellanos@fi.upm.es, Universidad Politécnica de Madrid, Spain

---

### Introduction

---

Software model transformation is one of the central activities in Model-Driven Development approach. Transforming allows to obtaining software model, i.e. UML diagram, of needed type for requirement analysis, designing, testing or other activity. One of the limitation of transformation tools is limited possibilities of composing transformation rules for those elements that are not linked directly [Kolovos, 2017], [QVT, 2016]. This fact becomes a motivation for authors to design framework for own tool to transform software models.

As requirement analysis is a very important step in software development process it is chosen to design a framework for Use Case to Communication diagrams transforming. A class diagram of this framework is represented in the figure 1.

---

### Description of the Proposed Framework

---

Information about software models objects is stored in the classes "Use\_Case" and "Collaboration" from "UML\_model" package.

Every type of Use Case Diagram notation is stored in separate list, for instance list of actors, precedents, and comments are created. Link between elements is characterized by two linked objects in UML diagram. Such an approach corresponds to graph metamodel representation, is proposed in book [Kolovos, 2017].

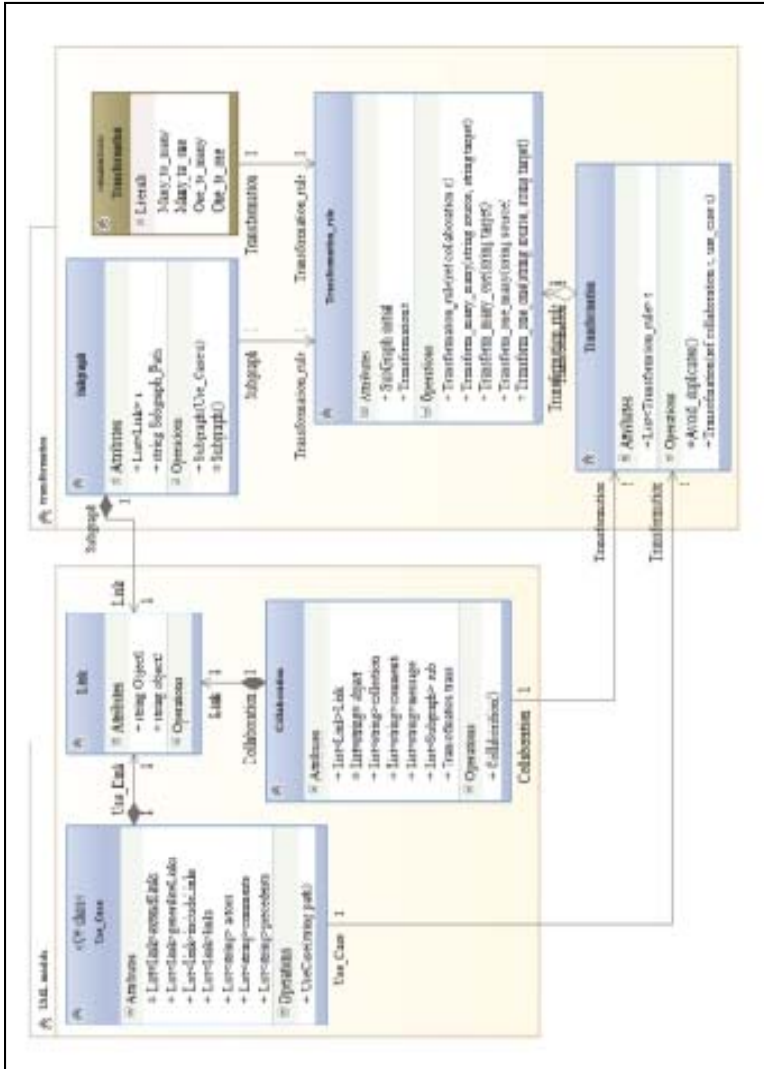


Figure 1. Class diagram of framework for transforming UML Use Case diagram to Communication one

Lists of Use Case diagram objects are populated in constructor *Use\_Case(string path)*. It obtains path to Use Case diagram, stored in XML based format [XMI, 2015]. Usually in this format environments for software models designing store information about UML diagrams. Examples of such environments are Microsoft Visual Studio, IBM Rational Software Architect, Integrated Development Environment Eclipse, ect.

Implementing LINQ queries XML tags can be proceeded. The aim of proceeding UML model file is to obtain lists of all its components and interconnections between them [Chebanyuk and Mironov, 2017].

The example of XML tags, corresponded to actor entity is given below.

```
<actor Id="f32af9db-669a-486d-a5a2-c66ebab0af85"  
  name="Actor1" isAbstract="false" isLeaf="false">
```

Then actor is linked with precedent "a":

```
<association Id="000dd500-e774-4754-b526-  
9fea979b34a1" isDerived="false"  
  sourceEndRoleName="Actor1" targetEndRoleName="a"  
  isLeaf="false" isAbstract="false">
```

Precedent itself is stored in the next XML string:

```
<memberEnd Id="9e1f21b1-9d2f-4e56-b61e-304d50b96f35"  
  name="a" isLeaf="false" isStatic="false"  
  isReadOnly="false" isDerived="false"  
  isDerivedUnion="false" aggregation="None"  
  isComposite="false">
```

The next step is to compose sub-graphs according to initial selection rules. Information about every sub-graph is stored in class "Subgraph" from "Transformation" package. Every sub-graph serves information to perform one transformation rule. Transformation rules are divided to several types, namely support one to one transformation, one to many, and many to many. Types of transformation rules are defined in enumeration "Transformation" from "Transformation" package.

Consider realization of different types of transformation rules.

Transformation rule one to one realized in the method *Transform\_one\_one(string source, string target)* in the next way: Every entity from list, named as source is added to list of entities, named target. Transformation many to one is realized by method *Transform\_many\_one(string target)*.

Entities, represented in target parameter are chosen from an initial sub-graph that is a property of Transformation\_rule class. Then, these entities are added to proper list of Collaboration Diagram.

Class "Transformation" stores list of transformation rules. Method *Transformation (ref collaboration c, use\_case u)* populates Collaboration Diagram entities. Collaboration Diagram is transmitted as a parameter to method.

And the last step is to delete duplicated entities from lists of collaboration diagram entities.

As a result of transformation, user obtains a list of sub-graphs and lists of Collaboration Diagram entities. They help user to verify designed collaboration diagram.

Information about software models links is stored in class "Link". Information about UML Collaboration diagram entities, obtained after transformation is stored in class "Collaboration".

---

## Conclusion

---

Detailed description of framework for UML Use Case to Communication Diagrams transformation is represented in this paper. Software developed in accordance to this framework allows to transform UML Use Case diagrams to Communication ones. It is important activity in requirement analysis process because after SCRUM meeting Use Case Diagrams are obtained, and communication diagrams are important software development artifacts that allow to perform data flow analysis. Such software serves as a gap for analyzing general functionality of future project (Use Case diagrams) and its detailed representation, considering data streams (Communication diagrams).

---

## References

---

[QVT, 2016] Meta Object Facility Query/View/Transformation, v1.3

<http://www.omg.org/cgi-bin/doc?formal/2016-06-03>

[XML, 2015] XML Metadata Interchange. Access mode <http://www.omg.org/spec/XMI/>

[Chebanyuk and Mironov, 2017] Chebanyuk Olena and Mironov Yuriy An approach of obtaining initial information for software models analysis. International Journal "Information content and Processing" Volume 4, Number 2, © 2017.

[Kolovos et. Al., 2017] Kolovos, D., Rose, L., Paige, R., & Garcia-Dominguez, A. (2017). The epsilon book. Structure, 178, 1-10. <https://www.eclipse.org/epsilon/doc/book/>

# APPLICATION OF GREY NUMBERS TO COMPUTATIONAL THINKING

**Michael Gr. Voskoglou**

Graduate Technological Educational Institute (TEI) of Western Greece, Greece  
mvosk@hol.gr

**Abstract:** *Computational thinking is a new problem solving method named so for its extensive use of computer science techniques. In the present paper Grey Numbers are used for evaluating student computational thinking skills. Grey numbers are indeterminate numbers which are defined with the help of the real intervals and they have found many applications in real life, science and engineering for handling approximate data.*

**Keywords:** *Grey Numbers (GNs), Computational Thinking (CT), Mathematical Modeling (MM), Whitenization, CT Quality*

---

## Introduction

---

*Computational thinking (CT)*, named so for its extensive use of computer science techniques, synthesizes *critical thinking* and *existing knowledge* and applies them to solve complex real world technological problems [Voskoglou & Buckley, 2012].

According to [Liu & Wang, 2010] CT is a hybrid of abstract thinking, logical thinking, constructive thinking and modeling thinking. The last one refers to the translation of objects or phenomena from the real world into mathematical equations (mathematical models) or computer relations (simulation models). One could claim that *modeling thinking* constitutes the essence of CT, since it synthesizes all the other modes of thinking for building up a model for the solution of the corresponding problem.

The paper in hands aims at using principles of the *Grey Systems* theory for evaluating student CT skills. In fact, when the student performance is evaluated with numerical grades, then the traditional way to assess the *mean performance* of a student class is the calculation of the average of all those grades. However, either for reasons of more elasticity or to comfort the teacher's existing uncertainty about the exact value of the numerical scores corresponding to each student's performance, frequently in practice

the assessment is made not by numerical scores but by *linguistic grades*, which makes the calculation of the average of them non feasible. Motivated by this we have used in earlier works tools of *Fuzzy Logic* for assessing the student performance; more explicitly the measurement of the existing uncertainty and *Fuzzy Numbers* [Voskoglou, 2017, Chapters 5 and 7]. In the present paper an alternative method will be developed by using *Grey Numbers (GNs)* instead of Fuzzy Numbers. Thus, after the introduction of the necessary for the understanding of the paper background on GNs, our new assessment method is developed and a classroom application is presented illustrating it. Finally, our conclusion is stated together with some hints for future research on the subject.

---

## Grey Numbers

---

A GN is an indeterminate number whose probable range is known, but which has unknown position within its boundaries. The GNs are defined with the help of the real intervals. More explicitly, if  $R$  denotes the set of real numbers, a GN, say  $A$ , can be expressed mathematically by

$$A \in [a, b] = \{x \in R : a \leq x \leq b\}.$$

If  $a = b$ , then  $A$  is called a *white number* and if  $A \in (-\infty, +\infty)$ , then it is called a *black number*. For general facts on GNs we refer to [Liu & Lin, 2010].

The well known arithmetic of the real intervals [Moore, et al., 1995] is used to define the basic arithmetic operations among the GNs. For example, if  $A \in [a_1, a_2]$  and  $B \in [b_1, b_2]$  are given GNs and  $k$  is a positive real number, one defines *addition* by  $A + B \in [a_1 + b_1, a_2 + b_2]$ , *scalar multiplication* by:

$$kA \in [ka_1, ka_2], \text{ etc.}$$

Let us denote by  $w(A)$  the white number with the highest probability to be the representative real value of the GN  $A \in [a, b]$ . The technique of determining the value of  $w(A)$  is called *whitenization* of  $A$ . One usually

defines  $w(A) = (1-t)a + tb$ , with  $t$  in  $[0,1]$ . This is known as the *equal weight whitening*. When the distribution of  $A$  is unknown, one takes  $t = \frac{1}{2}$ , which gives that  $w(A) = \frac{a+b}{2}$ .

---

## The Assessment Method

---

Let  $G$  be a group of  $n$  students participating in a certain activity (e.g. learning a new subject matter, problem-solving, etc.). Assume that one wants to estimate the mean performance of  $G$  in terms of the linguistic grades  $A = \text{Excellent}$ ,  $B = \text{Very Good}$ ,  $C = \text{Good}$ ,  $D = \text{Fair}$  and  $F = \text{Unsatisfactory}$ . For this, we introduce a numerical scale of scores from 0 to 100 and we assign these scores to the above linguistic grades as follows:  $A(100-85)$ ,  $B(84-75)$ ,  $C(74-60)$ ,  $D(59-50)$  and  $F(49-0)$ . This assignment, although it satisfies the common sense, it is not unique, depending on the user's personal goals.

We represent each linguistic grade by a GN, denoted with the same letter written in italic font. Namely, we introduce the GNs:  $A \in [85, 100]$ ,  $B \in [75, 84]$ ,  $C \in [60, 74]$ ,  $D \in [50, 59]$  and  $F \in [0, 49]$ . Let  $n_A$ ,  $n_B$ ,  $n_C$ ,  $n_D$  and  $n_F$  be the numbers of students whose performance was assessed by the grades  $A$ ,  $B$ ,  $C$ ,  $D$  and  $F$  respectively. Assigning to each student the corresponding GN we define the *mean value* of all those GNs to be the GN

$$M = \frac{1}{n} [n_A A + n_B B + n_C C + n_D D + n_F F]$$

Since  $n_A A \in [85n_A, 100n_A]$ ,  $n_B B \in [75n_B, 84n_B]$ ,  $n_C C \in [60n_C, 74n_C]$ ,  $n_D D \in [50n_D, 59n_D]$  and  $n_F F \in [0n_F, 49n_F]$ , we have that  $M \in [m_1, m_2]$ ,

with  $m_1 = \frac{1}{n} [85n_A + 75n_B + 60n_C + 50n_D + 0n_F]$  and

$$m_2 = \frac{1}{n} [100n_A + 84n_B + 74n_C + 59n_D + 49n_F].$$

It becomes evident that  $m_2$  is equal to the mean value of all student scores in the extreme case where the maximal possible numerical score

corresponds to each student for each linguistic grade (i.e. the  $n_A$  scores corresponding to A are 100, the  $n_B$  scores corresponding to B are 84, etc.). Also, in the opposite extreme case where the minimal possible numerical score corresponds to each student for each linguistic grade, the mean value of all those scores is equal to  $m_1$ . Therefore, since  $w(M) = \frac{m_1 + m_2}{2}$ , the use of GNs provides a reliable approximation of the group's mean performance.

**A Classroom Application:** The following Table depicts the performance of two student groups  $G_1$  and  $G_2$ , in a common mathematical test involving the solution of MM problems:

Table 1. Student performance

Grade	$G_1$	$G_2$
A	20	20
B	15	30
C	7	15
D	10	10
F	8	10
Total	60	85

Assigning to each student the corresponding GN we calculate the mean values  $M_1$  and  $M_2$  of all those GNs for the groups  $G_1$  and  $G_2$  respectively, which are approximately equal to

$$M_1 = \frac{1}{60} [20A + 15B + 7C + 10D + 8F] \in [62.42, 79.33] \text{ and}$$



$M_2 = \frac{1}{85} [20A + 30B + 15C + 10D + 10F] \in [62.42, 78.94]$ . Therefore

$$w(M_1) \approx \frac{62.42 + 79.33}{2} \approx 70.88 \quad \text{and} \quad w(M_2) \approx \frac{62.94 + 78.94}{2} \approx 70.94 .$$

Consequently both groups demonstrated a good (C) mean performance, with the mean performance of the second group being slightly better.

---

## Conclusion

---

In this paper using GNs we developed a method for assessing a student group mean performance, which is useful when the student performance is evaluated by qualitative grades. GNs play in general an important role in the everyday life for handling approximate data. Therefore, the development of further applications of them to real life problems is suggested as a promising area for future research.

---

## References

---

- [Liu & Lin, 2010] S. F. Liu, & Y. Lin (Eds.), *Advances in Grey System Research*, Berlin – Heidelberg: Springer, 2010.
- [Liu & Wang, 2010] Liu, J. & Wang, L., *Computational Thinking in Discrete Mathematics*, Proceedings of the IEEE 2nd International Workshop on Education Technology and Computer Science, pp. 413-416, 2010.
- [Moore, et al., 1995] R.A. Moore, R.B. Kearfort, & M.J. Clood, *Introduction to Interval Analysis*, 2nd Printing, Philadelphia: SIAM, 1995.
- [Voskoglou & Buckley, 2012] Voskoglou, M.Gr. & Buckley, S., *Problem Solving and Computers in a Learning Environment*, *Egyptian Computer Science Journal*, 36(4), 28-46, 2012.
- [Voskoglou, 2017] M. Gr. Voskoglou, *Finite Markov Chain and Fuzzy Logic Assessment Models: Emerging Research and Opportunities*, Columbia, SC, Createspace.com.– Amazon, 2017.
- [Alevizos et al., 2017] Alevizos, P.D., Theodorou, Y. & Vrahatis, M.N., *Correspondence Analysis with Grey Data: The Grey Eigen Value Problem*, *The Journal of Grey System*, 29(1), 92-105, 2017.
- [Voskoglou, 2015] Voskoglou, M. Gr., *Mathematical modelling as a teaching method of mathematics*, *Journal for Research in Innovative Teaching (National University, La Jolla, CA)*, 8(1), 35-50, 2015.

## **MODELING OF DATA TRANSITION PROCESS IN COMPUTER NETWORKS**

***E. Bezvershenko\**, *N. Guzii\*\**, *N. Procenko\*\*\**,**

\*Uzhhorod National University, Ukraine

\*\*National Aviation University, Ukraine (nn05@ukr.net)

\*\*\*National Aviation University, Ukraine

---

### **Introduction**

---

Systematic review of data transition process in computer networks allows to summarize existing knowledge and simplify further development of new network protocols, technologies of data transitions and other software engineering tasks [IEEE 802.2, 2011], [IEEE 802.3, 2012], [IEEE 802.11].

There are few papers addressed to systematize data about transition process in computer networks, for example [Chebanyuk, 2012].

But other aspects of computer networks functioning remain without proper attention. But today there are many data transitions standards that explain hardware functioning theoretically.

---

### **Contribution of this paper**

---

Absents of visual representation of computer hardware functioning become a motivation for authors to perform domain analysis operation and prepare visual representation of hardware functioning process. It is an initial information for successful further software maintains and software models transformation operation for architectural solutions designing.

Different operations of hub functioning are represented in Fig. 1.

A LAN switch is a device that typically consists of many ports that connect LAN segments (Ethernet and Token Ring) and a high-speed port (such as 100-Mbps Ethernet, Fiber Distributed Data Interface, or 155-Mbps ATM). The high-speed port, in turn, connects the LAN switch to other devices in the network.

When a LAN switch first starts up and as the devices that are connected to it request services from other devices, the switch builds a table that associates the MAC address of each local device with the port number through which that device is reachable.

That way, when Host A on Port 1 needs to transmit to Host B on Port 2, the LAN switch forwards frames from Port 1 to Port 2, thus sparing other hosts on Port 3 from responding to frames destined for Host B. If Host C needs to send data to Host D at the same time that Host A sends data to Host B, it can do so because the LAN switch can forward frames from Port 3 to Port 4 at the same time it forwards frames from Port 1 to Port 2.

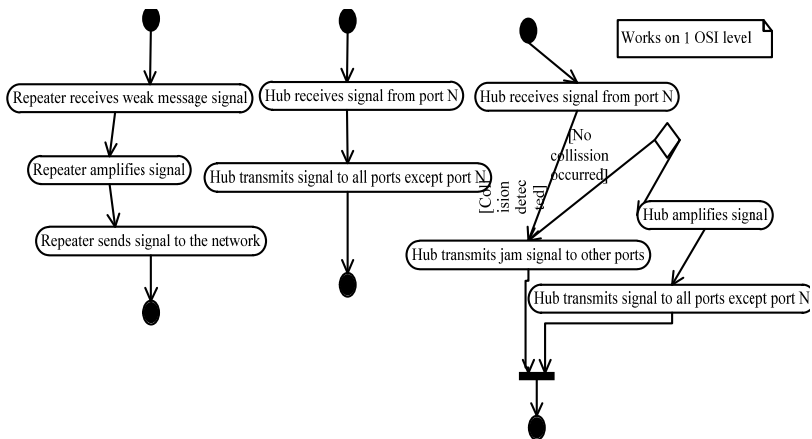


Fig. 1. Activity diagrams operation of a hub and a repeater functioning

Whenever a device connected to the LAN switch sends a packet to an address that is not in the LAN switch's table (for example, to a device that is beyond the LAN switch), or whenever the device sends a broadcast or multicast packet, the LAN switch sends the packet out all ports (except for the port from which the packet originated)-a technique known as flooding.

Because they work like traditional "transparent" bridges, LAN switches dissolve previously well-defined workgroup or department boundaries (Fig. 2). A network built and designed only with LAN switches appears as a flat network topology consisting of a single broadcast domain. Consequently, these networks are liable to suffer the problems inherent in flat (or bridged) networks-that is, they do not scale well. Note, however, that LAN switches that support VLANs are more scalable than traditional bridges.

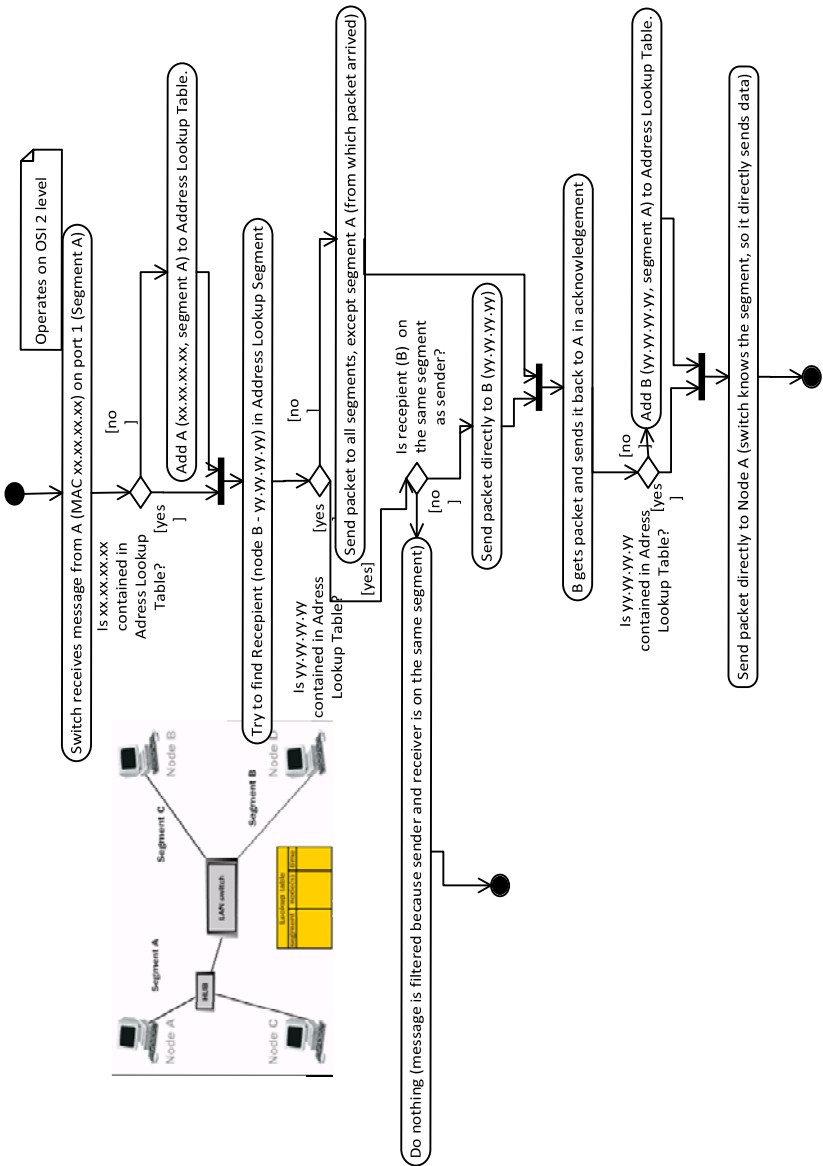


Fig. 2. Activity diagram: switch operation

---

## Conclusion

---

The visualization of data transition process in network hardware is performed in this paper. Designed behavioral software models fully correspond to rules of cognitive principles of convenient human understanding introduced in paper [Chebanyuk and Markov, 2015].

---

## References

---

[Chebanyuk, 2012] E. Chebanyuk [Designing of ontology for the application domain "Data transition processes in networks"](#). Інженерія програмного забезпечення. – 2012. – № 4(10). – С.24-33.

[Chebanyuk and Markov, 2015] E. Chebanyuk, Kr. Markov Software model cognitive value. International Journal "Information Theories and Applications", Vol. 22, Number 4, ITHEA 2015

[IEEE 802.2, 2011] IEEE 802.2 IEEE Computer Society. Standard for Information technology. Telecommunications and information exchange between systems. Local and metropolitan area networks. Specific requirements

[IEEE 802.3, 2012] IEEE 802.3, IEEE Computer Society. Ethernet

[IEEE 802.11, 2012] IEEE 802.11a, IEEE 802.11b, IEEE 802.11b, IEEE Computer Society.

## ФОРМУВАННЯ ДІАГРАМИ ВИКОНАННЯ ЗАВДАНЬ ЗА МЕТОДОЛОГІЄЮ AGILE

*Жмаєва Ю.В., Удовенко С.Г., Чала Л.Е*

ХНУРЕ, larysa.chala@nure.ua

Останнім часом отримали розповсюдження методи розрахунку та прогнозування оптимальної кількості ресурсів для виконання ІТ проектів за методологією Agile. В цих методах мають бути враховані як стандартні показники, що характеризують людські ресурси, так і додаткові показники, які дозволяють підвищити наочність представлення результатів (фокус-фактор, кількість незапланованого обсягу робіт, кількість роботи, що залишилася, за тижнями, відмінність оцінок завдань) [K Rubin, 2012]. Для динамічного формування діаграми виконання завдань за методологією Agile був розроблений Web-додаток на основі схеми використання шаблонів MVC, який дозволяє розділяти дані, подання та обробку дій користувача на три окремих компоненти: модель, представлення, вигляд [Жмаєва, 2012]. До створеної ітерації користувач може додати список проектних задач, який містить характеристики кожного запису. В ході розробки для реалізації MVC були спроектовані групи класів: сервіси, калькулятори, сутності, утильні класи, класи-репозиторії і контролери. Використання спроектованих груп класів дає можливість відображення об'єктно-орієнтованої моделі даних в традиційні реляційні бази даних. Ядром програмної реалізації запропонованої технології є база даних, яка зберігає інформацію, необхідну для побудови діаграми прогресу розробки. Вона містить інформацію про ітерації, які створюють репозиторій для прогнозування незапланованого обсягу завдань і похибки в оцінці завдань.

---

### Література

[K Rubin, 2012] Rubin, K. S. Essential Scrum: A Practical Guide to the Most Popular Agile Process / Rubin, K. S. – Н.: Addison Professional, 2012. – 281 p.

[Жмаєва Ю.В., 2012] Адаптивное прогнозирование оптимального количества ресурсов ИТ проекта по методологии Agile [Текст] / Ю.В. Жмаєва, Л.Э. Чала, С.Г. Удовенко// АСУ и приборы автоматизи. – 2015. – № 173. – С. 4–13.

## MODERN LANGUAGES AND TOOLS FOR RESTFUL WEB API MODELLING

*Kyrylo Malakhov*

V.M. Glushkov Institute of Cybernetics, [malakhovks@outlook.com](mailto:malakhovks@outlook.com)

The given paper represents an overview of modern RESTful API Description Languages (OAS, RAML, API Blueprint) and brief comparison of current RESTful web API modelling tools. REST was coined by Roy Fielding in his Ph.D. dissertation [Fielding R.T. and Taylor R.N., 2000] to describe a design pattern for implementing networked systems. REST is Representational State Transfer, an architectural style for designing distributed systems. RESTful API DLs are formal languages designed to provide a structured description of a RESTful web API that is useful both to a human and for automated machine processing.

Table 1. Comparison of RESTful API DLs.

DLs	Format	Code generation	Open Source	Dev tools	Design	References
RAML	YAML	yes	yes	Good	API first	<a href="https://raml.org/">https://raml.org/</a>
API Blueprint	Mark-down	limited	yes	Limited	Design first	<a href="https://apiblueprint.org/">https://apiblueprint.org/</a>
OAS	JSON	yes	yes	Good	Existing API	<a href="https://swagger.io/">https://swagger.io/</a>

OAS has a very strong modeling language for defining exactly what's expected of the system – very useful for testing and creating coding stubs for a set of APIs. RAML is designed to support a design-first development flow and focuses on consistency. Apiary blueprint is more documentation-focused, with user-readable models and documentation as its priority. Each project brings different strengths and weaknesses to the table, and in the end it's about what strengths you need and which weaknesses you cannot afford. Overall, RAML fared the best in these different categories and, while the developer community is not as large as the others, it's safe to say it will keep growing.

---

### References

[Fielding R.T. and Taylor R.N., 2000] Fielding R.T., Taylor R.N. Architectural styles and the design of network-based software architectures. 2000.

## ПОРІВНЯННЯ РІЗНИХ ТЕХНОЛОГІЙ СТВОРЕННЯ І ВИКОРИСТАННЯ WEB-СЕРВІСІВ

*Міловідов Ю.О.*

НУБіП України, [milovidov@email.ua](mailto:milovidov@email.ua)

На сьогоднішній день існує велика кількість некомерційних і комерційних засобів розробки веб-сервісів. Всі вони розрізняються за ступенем повноти, можливостям, умовам ліцензування, вимогам до програмно-апаратного забезпечення і т.п. Мета дослідження: на основі висновків порівняння різних технологій зі створення і використання web-сервісів, надати корисну інформацію розробникам, які бажають реалізувати і розгорнути свої web-сервіси. Озброївшись цією інформацією, можна зрозуміти яку технологію краще використовувати.

Проведено тестування часу web-сервісів, створених за технологією .Net і Java. Час відгуку практично однаковий.

Проведено 100 випробувань і обчислено середній час роботи web-сервісу, що повертає записи з однієї таблиці БД (близько 30000 записів). Axis помітно поступається за часом виконання web-сервісу. IIS працює у два рази швидше.

Проведена спроба зробити тисячі одночасних підключень до web-сервісу. Даний тест показує, що Apache Axis більш стійкий до багатьох підключень.

Порівнюючи технології .Net і Java щодо створення і використання web-сервісів можна зробити наступні висновки:

Web-сервіс, створений за допомогою .Net і розгорнутий на web-сервері IIS працює швидше, за менший час повертає дані, ніж web-сервіс, реалізований за допомогою Apache Axis. Але web-сервіс на IIS програє по стійкості до багатьох підключень.



## КЛЮЧОВІ АСПЕКТИ ВЗАЄМОДІЇ ВЕБ-СЕРВІСІВ

**Скалова В.А.**

НАУ, valentine.skalova@livenau.net

Дослідження ключових аспектів взаємодії веб-сервісів базується на описі стандартів, що дозволяють зберігати інформацію про сервіси та механізмів їх взаємодії.

Ключові стандарти, що дозволяють зберігати інформацію про веб-сервіси:

- ✓ WSDL використовується, щоб описати функції сервісу: його точку входу, вхідні параметри і результат.
- ✓ XML використовується для передачі значень параметрів і результатів роботи сервісу.
- ✓ SOAP використовується як протокол обміну XML файлами через мережу.
- ✓ HTTP використовується в якості мережевого протоколу.

Схема взаємодії сервісу та застосунку представлена на рис 1.

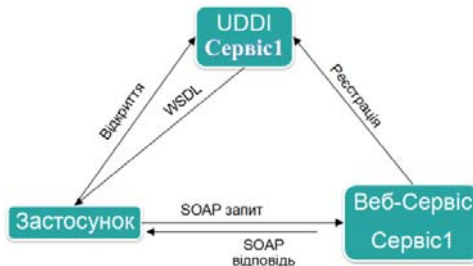


Рис 1. Схема взаємодії сервісу та застосунку

Механізм взаємодії складається з наступних кроків:

- ✓ Веб-сервіс реєструється шині (протокол HTTP).
- ✓ Застосунок звертається до шини із запитом та отримує WSDL (на мові XML) опис функціоналу сервісу.
- ✓ Після цього застосунок має прямий доступ до сервісу.
- ✓ Дані передаються за допомогою SOAP повідомлень (запити та відповіді, мова XML)

## МЕТОДИКА РОЗРАХУНКУ ПОКРИТТЯ КОДУ ТЕСТАМИ

*Дишлевий О.П.*

НАУ, oleksiy.dyshlevyy@livenau.net

Актуальна на сьогодні проблема – ризики, які виникають унаслідок неякісного програмного забезпечення (ПЗ) [Tharwon Arnuphaptrairong , 2011]. Більш того не всі проекти з розробки програмного забезпечення (ПЗ) є успішними [Janusz Laski nad William Stanley , 2009]. Одна з головних причин їх провалу – недостатня ефективність тестування продукту [B. W. Boehm et. all, 1978].

Для оцінки якості коду багатьох проектів зараз використовують метрики покриття коду (МПК). Найчастіше використовуються наступні метрики: покриття рядків, покриття гілок, покриття шляхів. Існуючі підходи та засоби вимірювання часто не дають цілісної картини про реальний стан коду [Janusz Laski and William Stanley , 2009].

Для вирішення проблеми пропонується розраховувати зведену метрику загальне покриття (TC) на основі покриття рядків (SC), покриття гілок (BC) та покриття шляхів (PC):

$$TC = a * SC + b * BC + c * PC$$

де  $a$ ,  $b$ ,  $c$  – вагові коефіцієнти, при чому  $a + b + c = 1$ ; SC – SC у відсотках; BC – BC у відсотках; PC – PC у відсотках.

На основі аналізу результатів вимірювань java проектів виявлено, що значення PC – 25% від значення SC, а значення BC становить 80% від SC. Як результат отримуємо  $a = 0,16$ ,  $b = 0,2$ ,  $c = 0,64$ .

---

### Література

[Tharwon Arnuphaptrairong, 2011] Tharwon Arnuphaptrairong. Top Ten Lists of Software Project Risks: Evidence from the Literature Survey. IMECS International MultyConference of Engineers and Computer Scientistis. VOL 1, March 2011.-732-737pp.

[B. W. Boehm et. all, 1978] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. MacLeod, and M. J. Merritt. Characteristics of Software Quality. North Holland Publishing Company, 1978.-524p.

[Janusz Laski and William Stanley , 2009]. Janusz Laski, William Stanley. Software Verification and Analysis. An Integrated, Hands-On Approach - Springer-Verlag London Limited, 2009.-205p.

## КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА НАВЧАЛЬНОЇ ЛАБОРАТОРІЇ З RICH КЛІЄНТАМИ

**Сокорчук І. П.**

ХНУРЕ, [ihor.sokorchuk@nure.ua](mailto:ihor.sokorchuk@nure.ua)

Для навчальних лабораторій може бути застосована система побудована за клієнт-серверною мережевою архітектурою із товстим (Rich, Fat) клієнтом на базі GNU/Linux:

Для робочих місць використовуються бездисккові робочі станції із збільшеним обсягом оперативної пам'яті, що підтримують завантаження із мережі за протоколом PXE.

У локальній мережі встановлюється один або кілька серверів, що надають клієнтам такі сервіси:

- ✓ завантаження на робочі станції завантажувача операційної системи (Boot Loader) – dhcpd, tftpd, pxelinux;
- ✓ завантаження на робочі станції ядра операційної системи (vmlinuz) та кореневого образу файлової системи (initrd або initramfs) із базовим системним ПЗ – tftpd;
- ✓ налаштування операційної системи на робочих станціях – dhcpd, клієнтські скрипти bash;
- ✓ завантаження робочого образу файлової системи із основним ПЗ – ftpd, httpd;
- ✓ доступ до додаткового ПЗ та даних – nfsd;
- ✓ доступ до централізованого сховища даних – sshd, fuse.

Таке рішення має низку переваг, зокрема: централізоване адміністрування програмного комплексу, спрощене обслуговування ПЗ на робочих місцях, централізоване збереження інформації, порівняно невелике навантаження на мережу, менші вимоги до обчислювальних ресурсів серверів, краще використання ресурсів робочих станцій,

Описане рішення використано для навчальної лабораторії кафедри програмної інженерії ХНУРЕ.

# МЕТОДЫ ОЦЕНКИ КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ

*Клюев Е.И., Гриненко Е.А., Отрошенко В.Г.*

Чаще всего для трактовки качества программных средств (ПС) используется определение ISO 9001, согласно которому качество есть «степень соответствия присущих характеристик требованиям». Качество кода может определяться различными критериями. Увеличение сложности и размеров современных комплексов программ при одновременном росте ответственности выполняемых функций повысило требования со стороны заказчиков и пользователей к их качеству и безопасности применения. Испытанным средством обеспечения высокой эффективности и качества функционирования программ и программных комплексов являются международные стандарты. В настоящее время недостаточно формализованы механизмы формирования значений, определяющих качество программного продукта, как их следует измерять, сравнивать с требованиями, отраженными в контракте, техническом задании или спецификациях. Нечеткое декларирование в документах требуемых значений характеристик качества программных средств вызывает конфликты между заказчиками и разработчиками из-за разной трактовки одних и тех же характеристик. В связи с этим стратегической задачей в жизненном цикле современных информационных систем стало обеспечение требуемого качества программных средств и баз данных. Исходными данными и высшим приоритетом при выборе показателей качества в большинстве случаев являются назначение, функции и функциональная пригодность соответствующего программного средства. Оценка качества ПС проводится на фазах жизненного цикла (Гост 28195-89, табл. 1) и включает выбор номенклатуры показателей, их оценку и сопоставление значений показателей, полученных в результате сравнения с базовыми значениями. Общая оценка качества ПС формируется экспертами на основании полученных значений оценок фактора качества. При оценке

качества ПС методом экспертного опроса составляется таблица значений базовых показателей качества ПС. В стандарте Гост 28195-89 отмечается, что необходимые данные для оценки уровня разработки программного продукта могут быть сформированы путем использования экспертного метода, поэтому качество показателя определяется как усредненная оценка экспертных высказываний. Для повышения эффективности экспертного метода предлагается его модернизация за счет использования аппарата нечеткой логики. На основе аппарата теории нечеткой логики разработана процедура принятия решений в расплывчатых условиях, которая может быть использована при определении значений оценочных элементов ПС и которые можно рассматривать как требования заказчика к характеристикам проектируемого программного обеспечения. Данные, полученные на основе модернизированного экспертного метода, могут быть использованы в качестве эталонных значений при обучении нейронной сети (ГОСТ 28195-89). Обученная нейронная сеть используется в дальнейшем для оценки уровня разработки ПЗ путем сравнения эталонных значений с фактическими данными. Выгрузка данных из нейронной сети в офисное приложение Excel (пакет Статистика) может быть использовано для расчета необходимых показателей оценки качества ПЗ в соответствии с методикой Гост 28195-89. В таблице 1 приведены некоторые данные, используемые для оценки качества разработки ПС пакетом «Статистика». В качестве исходных данных выступают оценочные элементы.

Таблица 1 Описание оценочных элементов

<b>Оценочный элемент</b>
Наличие требований к программе по устойчивости функционирования при наличии ошибок во входных данных
Возможность обработки ошибочных ситуаций
Полнота обработки ошибочных ситуаций

На рис.1-2 демонстрируются таблицы, раскрывающие основные положения использования пакета «Статистика» при решении задачи оценки уровня разработки ПС.

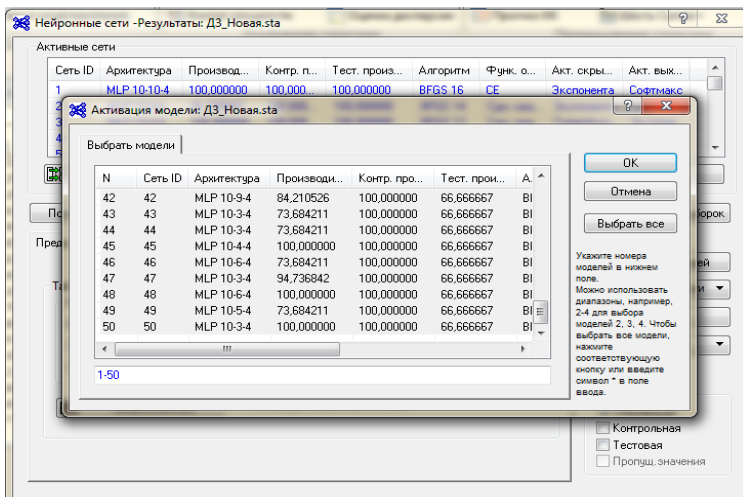
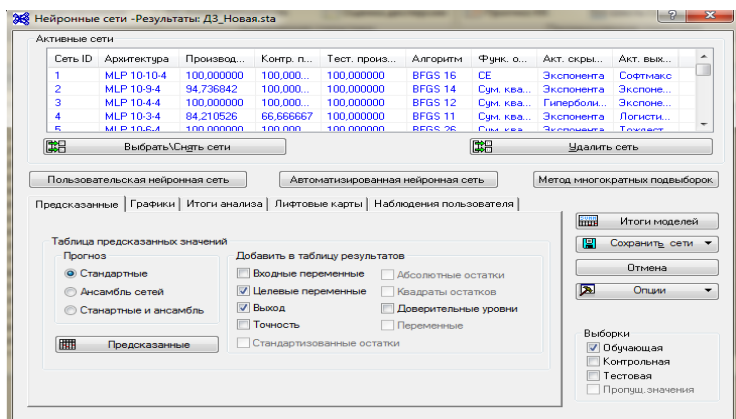


Рис 1. Результаты обучения нейронной сети

Таблица предсказанных значений для Результ Выборки: Тестовая		
Наблюд. номер #	Результат Целевая	Результат - Выход 1. MLP 10-10-4
1	удовлетворительно	неудовлетворительно
2	неудовлетворительно	неудовлетворительно
3	удовлетворительно	удовлетворительно
4	хорошо	хорошо
5	отлично	отлично
6	отлично	отлично
7	удовлетворительно	удовлетворительно
8	хорошо	отлично
9	неудовлетворительно	неудовлетворительно

	1	2	3	4	5	6	7	8	9	10	Наблюд. номер #	Результат Целевая	Результат - Выход 1. MLP 10-10-4
	HO101	HO102	HO103	HO104	HO105	HO106	HO107	HO108	HO109	HO110			
1	0,56	0,53	0,51	0,56	0,37	0,88	0,18	0,21	0,10	0,10	1	удовлетворительно	неудовлетворительно
2	0,17	0,29	0,16	0,26	0,18	0,27	0,51	0,18	0,18	0,90	2	неудовлетворительно	неудовлетворительно
3	0,35	0,20	0,56	0,10	0,21	0,90	0,07	0,96	0,61	0,66	3	удовлетворительно	удовлетворительно
4	0,41	0,65	0,36	0,98	0,99	0,86	0,99	0,66	0,70	0,61	4	хорошо	хорошо
5	0,94	0,90	0,82	0,70	0,88	0,52	0,83	0,80	0,95	0,80	5	отлично	отлично
6	0,32	0,89	0,93	0,97	0,90	0,96	0,44	0,75	0,84	0,96	6	отлично	отлично
7	0,56	0,81	0,35	0,33	0,20	0,51	0,59	0,35	0,48	0,23	7	удовлетворительно	удовлетворительно
8	0,79	0,98	0,99	0,75	0,92	0,72	0,69	0,98	0,68	0,78	8	хорошо	отлично
9	0,12	0,28	0,65	0,61	0,17	0,18	0,30	0,57	0,08	0,23	9	неудовлетворительно	неудовлетворительно
10	0,98	0,70	0,44	0,76	0,56	0,93	0,85	0,87	0,39	0,69	10	хорошо	хорошо
11	0,54	0,88	0,75	0,29	0,25	0,85	0,66	0,78	0,99	0,70	11	хорошо	хорошо
12	0,04	0,97	0,19	0,87	0,19	0,25	0,14	0,19	0,27	0,17	12	отлично	неудовлетворительно
13	0,30	0,30	0,01	0,10	0,57	0,62	0,25	0,30	0,38	0,15	13	неудовлетворительно	неудовлетворительно
14	0,10	0,21	0,72	0,40	0,17	0,15	0,67	0,19	0,10	0,20	14	неудовлетворительно	неудовлетворительно
15	0,50	0,64	0,33	0,97	0,93	0,81	0,29	0,88	0,85	0,87	15	хорошо	хорошо
16	0,62	0,69	0,98	0,32	0,64	0,90	0,12	0,23	0,27	0,04	16	удовлетворительно	удовлетворительно
17	0,09	0,07	0,18	0,21	0,59	0,32	0,17	0,87	0,09	0,24	17	неудовлетворительно	неудовлетворительно
18	0,79	0,81	0,95	0,89	0,73	0,69	0,89	0,94	0,65	0,94	18	отлично	отлично
19	0,10	0,40	0,27	0,27	0,10	0,28	0,27	0,16	0,87	0,22	19	неудовлетворительно	неудовлетворительно
20	0,96	0,71	0,99	0,22	0,66	0,76	0,84	0,73	0,48	0,75	20	хорошо	хорошо
21	0,35	0,64	0,66	0,66	0,56	0,66	0,20	0,06	0,68	0,21	21	удовлетворительно	удовлетворительно
22	0,91	0,92	0,98	0,84	0,58	0,83	0,75	0,95	0,56	0,88	22	отлично	отлично
23	0,82	0,83	0,98	0,72	0,85	0,65	0,97	0,73	0,78	0,99	23	отлично	отлично
24	0,98	0,27	0,91	0,10	0,54	0,48	0,93	0,76	0,86	0,96	24	отлично	отлично

Рис 2. Результаты предсказания и оценки уровня разработки ПС.

Для расчета комплексного показателя оценки качества разработки ПС (Гост 28195-89), сформированные в пакете «Статистика» результирующие данные выгружаются в офисное приложение Excel (рис.3).

0,56	0,53	0,51	0,08	0,37	0,08	0,18	0,21	0,10	0,10	неудовлетворительно	0,27
0,17	0,29	0,16	0,26	0,18	0,27	0,51	0,18	0,18	0,90	неудовлетворительно	0,31
0,35	0,20	0,56	0,10	0,21	0,50	0,07	0,96	0,61	0,66	удовлетворительно	0,42
0,41	0,65	0,36	0,98	0,99	0,86	0,99	0,66	0,70	0,61	хорошо	0,72
0,94	0,90	0,82	0,70	0,88	0,52	0,83	0,80	0,95	0,80	отлично	0,81
0,92	0,89	0,95	0,97	0,90	0,96	0,44	0,75	0,84	0,98	отлично	0,86
0,56	0,81	0,35	0,33	0,20	0,51	0,59	0,35	0,48	0,23	удовлетворительно	0,44
0,79	0,98	0,99	0,75	0,92	0,72	0,69	0,98	0,68	0,78	отлично	0,83
0,12	0,26	0,65	0,61	0,17	0,18	0,30	0,57	0,08	0,23	неудовлетворительно	0,32
0,98	0,70	0,44	0,76	0,56	0,93	0,85	0,87	0,39	0,69	хорошо	0,72
0,54	0,88	0,75	0,29	0,25	0,85	0,66	0,78	0,99	0,70	хорошо	0,67
0,04	0,97	0,19	0,87	0,19	0,25	0,14	0,19	0,27	0,17	неудовлетворительно	0,33
0,30	0,30	0,01	0,10	0,57	0,62	0,25	0,30	0,38	0,15	неудовлетворительно	0,30
0,10	0,21	0,72	0,40	0,17	0,15	0,67	0,19	0,10	0,20	неудовлетворительно	0,29
0,50	0,84	0,35	0,97	0,95	0,81	0,25	0,88	0,65	0,87	хорошо	0,71
0,62	0,69	0,98	0,32	0,64	0,90	0,12	0,23	0,27	0,04	удовлетворительно	0,48
0,09	0,07	0,18	0,21	0,59	0,32	0,17	0,87	0,09	0,24	неудовлетворительно	0,28
0,79	0,81	0,95	0,89	0,73	0,69	0,89	0,94	0,65	0,94	отлично	0,83
0,10	0,40	0,27	0,27	0,10	0,28	0,27	0,16	0,87	0,22	неудовлетворительно	0,29
0,96	0,71	0,99	0,22	0,66	0,76	0,84	0,73	0,48	0,75	хорошо	0,71
0,35	0,64	0,66	0,66	0,56	0,66	0,20	0,06	0,66	0,21	удовлетворительно	0,47
0,91	0,92	0,98	0,84	0,58	0,83	0,75	0,95	0,56	0,88	отлично	0,82
0,82	0,83	0,98	0,72	0,85	0,65	0,97	0,73	0,78	0,99	отлично	0,83
0,96	0,27	0,91	0,10	0,54	0,46	0,93	0,76	0,86	0,98	хорошо	0,68
0,41	0,09	0,69	0,97	0,82	0,15	0,12	0,55	0,93	0,07	удовлетворительно	0,48

	количество	сумма	ср.арифм.
неудовлетворительно	8	2,391140909	0,298892614
удовлетворительно	5	2,292094134	0,458418827
хорошо	6	4,203369926	0,700561654
отлично	6	4,981746759	0,830291126
фактор	0,572041055		

Рис 3. Результаты формирования комплексного показателя качества разработки ПС.