

Literatura

1. *Batishchev D.I., L'vovich Ya.E., Frolov V.N.* Optimizaciya v SAPR. Voronezh: Izd-vo Voronezhskogo gosuniversiteta, 1997. 416 s.
2. *Bolgova M.A., Evdokimova E.A.* Prinyatie upravlencheskih reshenij v usloviyah transformacii vysshego obrazovaniya // Vestnik universiteta. 2016. № 3. S. 195–197.
3. *Bolgova M.A., Podlegaev A.V.* Transformaciya vysshego obrazovaniya v ramkah sozdaniya opornyh universitetov: social'no-ekonomicheskij analiz // Vestnik Associacii vuzov turizma i servisa. 2016. T. 10, № 1. S. 44–50.
4. *Ivashkevich V.B.* Organizaciya upravlencheskogo ucheta po centram otvetstvennosti i mestam formirovaniya zatrat // Buhgalterskij uchet. 2000. № 5. S. 56–59.
5. *Kryuchin O.V.* Prognozirovaniye vremennyh ryadov s pomoshch'yu iskusstvennyh nejronnyh setej i regressivnyh modelej na primere prognozirovaniya kotirovok valyutnyh par // Issledovaniya v Rossii. 2010. № 30. S. 354–362.
6. *L'vovich I.Ya., L'vovich Ya.E., Frolov V.N.* Informacionnye tekhnologii modelirovaniya i optimizacii: kratkaya teoriya i prilozheniya. Voronezh: Nauchnaya kniga, 2016. 444 s.
7. *L'vovich Ya.E.* Mnogoal'ternativnaya optimizaciya: teoriya i prilozheniya. Voronezh: Kvarta, 2006. 428 s.
8. *L'vovich Ya.E., Mihel' A.A.* Strukturizaciya upravleniya resursnymi i ob'emnymi karakteristikami sistemy vysshego obrazovaniya s ispol'zovaniem sredstv optimal'nogo vybora // Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta. 2014. T. 10, № 4. S. 13–16.
9. *Kashirina I.L., L'vovich Ya.E., Sorokin S.O.* Integral'noe ocenivaniye effektivnosti setevykh sistem s klasternoj strukturoj // Ekonomika i menezhment sistem upravleniya. 2015. № 1-3 (15). S. 330–337.
10. *Tarhov D.A.* Nejrosetevye modeli i algoritmy: spravochnik. M.: Radiotekhnika, 2014. 352 s.
11. *Donges N.A* Complete Guide to the Random Forest Algorithm. URL: <http://builtin.com/dato-science/random-forest-algorithm> (data obrashcheniya: 11.09.2020).

DOI: 10.25586/RNU.V9187.20.04.P056

УДК 004.054

А.П. Киреев, С.А. Шаров

ПРОГРАММНЫЙ КОМПОНЕНТ РАСЧЕТА ПОКАЗАТЕЛЕЙ
КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ

Представлены результаты исследований по обоснованию и разработке набора метрик, с помощью которых можно рассчитать и оценить качество программных средств специального назначения (ПССН) автоматизированных систем управления (АСУ), разработанных с помощью объектно ориентированных языков программирования (ООЯП). Представлен алгоритм расчета объектно ориентированных метрик (ООМ) ПССН.

Ключевые слова: программное обеспечение, программные средства специального назначения, показатели качества, объектно ориентированные метрики, программные компоненты.

A.P. Kireev, S.A. Sharov

PROGRAM COMPONENT FOR CALCULATION
OF INDICATORS QUALITY OF SOFTWARE

The article presents the results of studies on the justification and development of a set of metrics with which you can calculate and evaluate the quality of special-purpose software tools (SPST) of automated control systems (ACS) developed using object-oriented programming languages (OOPL). The algorithm for calculating the object-oriented metrics (OOM) of PSSN is presented.

Keywords: software, special-purpose software, quality indicators, object-oriented metrics, software components.

Введение

Качество программного обеспечения (ПО) – способность программного продукта удовлетворять заявленные и подразумеваемые потребности при использовании в определенных условиях. Качество программного обеспечения описывается и оценивается посредством определения, измерения и сравнения его существенных характеристик с заданными требованиями или неким эталоном. Таким образом, требования к качеству ПО представляют собой совокупность утверждений относительно значений его характеристик и условий применения.

Проектирование программного обеспечения для АСУ [5] требует применения новейших технологий разработки программных средств. Для адекватной оценки качества PSSN необходимы программные компоненты, базирующиеся на объектно ориентированных метриках оценки качества программного обеспечения (ПО).

Модель процесса оценки качества программного продукта (ПП) согласно ГОСТ Р ИСО/МЭК 25040–2014 [2] можно представить в виде алгоритма (рис. 1), состоящего из следующих задач:

- спецификация целей и требований к оценке, идентификация частей ПО, подлежащих оценке, определение строгости оценки;
- выбор показателей качества (ПК), определение критериев принятия решения для них, определение критериев принятия решения для оценки;
- планирование деятельности по оценке;
- выполнение измерений и применение критериев для ПК и оценки;
- анализ результатов оценки и оформление результатов.

В настоящее время при разработке PSSN чаще всего используются ООЯП [6]. Это обстоятельство обусловлено тем, что ООЯП, являясь результатом поступательного развития технологий программной инженерии [8], позволяют создавать более эффективные и надежные [7], лучше понимаемые и легче сопровождаемые и перенастраиваемые программные системы [3].

Концептуальной базой объектно ориентированного программирования является объектная модель, которая базируется на следующих основных принципах и методах: абстрагирование, инкапсуляция, модульность, иерархия [2].

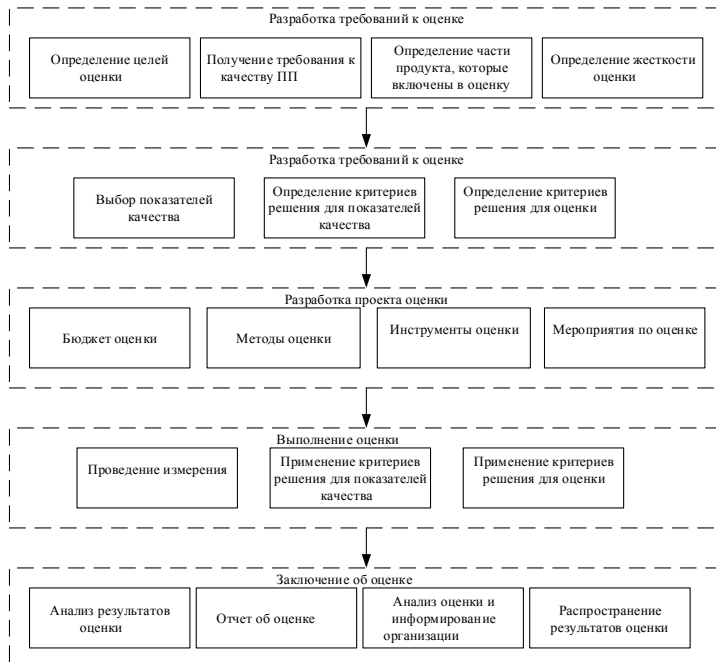


Рис. 1. Алгоритм оценки качества ПО

Абстрагирование – способ выделить значимые характеристики объекта, отличающие его от других типов объектов и четко определяющие его концептуальные границы [6]. Выделение полного и достаточного набора абстракций является главной задачей объектно ориентированного проектирования ПССН [9].

Инкапсуляция – свойство, позволяющее объединить данные и методы, работающие с ними, в класс и скрыть детали реализации от пользователя.

Рациональная инкапсуляция позволяет локализовать и скрыть части реализации системы, которые могут подвергнуться изменениям [1].

Принцип модульности служит для преодоления проблем, связанных с размерностью и сложностью разрабатываемой системы [4, 5]. Разбиение ПССН на модули (классы и пакеты) позволяет не только бороться со сложностью, но и заставляет тщательно продумывать и хорошо документировать интерфейсы между модулями, что, в свою очередь, облегчает процесс объектной декомпозиции системы. Правильное разбиение программы на модули, определение связей и зависимостей между ними является такой же важной задачей, как и задача построения оптимального набора абстракций [10]. Вероятность внесения изменений в интерфейсные элементы модулей должна быть сведена к минимуму [13]. Это достигается уменьшением сцепления между модулями ПС [16].

Иерархия – это упорядочивание абстракций путем разложения их по уровням [14]. В объектно ориентированных системах используются два типа иерархических структур: иерархия классов (is-a), иерархия объектов (part of) [11].

Без строгого следования описанным выше принципам возможности ООЯП будут потеряны или искажены, а их использование не по назначению приведет к ухудшению харак-

Киреев А.П., Шаров С.А. Программный компонент расчета показателей...

теристик разрабатываемых ПССН [12]. Для обеспечения максимально полного и эффективного использования преимуществ [18] и возможностей ООЯП в разрабатываемых на их основе ПССН необходимо предоставить проектировщикам и программистам средства [15], с помощью которых они смогли бы измерить и оценить качество ПССН с точки зрения реализации в них указанных выше принципов и возможностей [14].

Для любого инженерного продукта метрики должны ориентироваться на его уникальные характеристики. С точки зрения метрик выделяют пять характеристик объектно ориентированных систем: локализация, инкапсуляция, информационная закрытость, наследование и способы абстрагирования объектов. Эти характеристики оказывают максимальное влияние на объектно ориентированные метрики.

Обоснование состава объектно ориентированных метрик

Для оценки качества программных средств (ПС), созданных при помощи языков объектно ориентированного программирования, используется класс объектно ориентированных метрик. В данном классе наиболее используемыми являются: набор метрик Мартина, метрики М. Лоренца и Д. Кидда [13], набор метрик Чидамбера и Кемерера [14] и набор метрик MOOD (Metrics of ObjectOriented Design) Ф. Абреу [9].

Объектно ориентированные метрики ПССН должны удовлетворять следующим требованиям:

- ООМ должны покрывать все базовые механизмы объектно ориентированной парадигмы – инкапсуляцию, наследование, полиморфизм, посылку сообщений;
- алгоритмы расчета ООМ должны использовать объективные характеристики и свойства программ и выдавать количественные значения метрик;
- метрики должны быть независимы от размера оцениваемого программного продукта и от языка программирования, на котором написан оцениваемый продукт.

Как показал проведенный анализ [2, 3, 6, 14, 16], для оценки качества объектно ориентированных программ целесообразно использовать следующий набор метрик:

- метрики оценки статистических характеристик программного кода (количество строк в программном проекте, количество строк кода классов, количество строк кода метода класса, количество комментариев);
- оценки системы классов с точки зрения реализации в ней базовых принципов объектно ориентированного программирования (ООП) (глубина наследования, количество потомков, количество методов объекта, количество унаследованных методов, количество закрытых методов, полиморфизм, сцепление и связность классов);
- оценки качества распределения и локализации классов в системе пакетов (количество классов в пакете, скрытые классы, количество методов в пакете, нестабильность пакета, абстрактность пакета, центробежное сцепление, контекстная связанность пакета).

Наиболее важными при проектировании ПССН являются следующие ООМ:

- MWF (Weighted methods per class) – фактор закрытости метода;
- AHF (Attribute Hiding Factor) – фактор закрытости атрибута;
- MIF (Method Inheritance Factor) – фактор наследования метода;
- AIF (Attribute Inheritance Factor) – фактор наследования атрибута;
- POF (Polymorphism Factor) – фактор полиморфизма;
- COF (Coupling Factor) – фактор сцепления;

- NCP (Number of Classes in Package) – количество классов в пакете;
- WMC (Weighted methods per class) – сложность методов класса;
- CBO (Coupling between object classes) – сцепление между классами.

С помощью этих метрик можно оценить, насколько полно и точно в проекте используются такие базовые механизмы объектно ориентированной парадигмы, как инкапсуляция, наследование, полиморфизм, абстракция, обмен сообщениями и др. Каждая из этих метрик относится к основному механизму объектно ориентированной парадигмы – инкапсуляции (МНФ и АНФ), наследованию (МІФ и АІФ), полиморфизму (РОФ) и посылке сообщений (СОФ).

Перечень и описание способов расчета некоторых ООМ, а также критерии оценки качества ПССН представлены в таблице 1.

По наблюдениям [11]:

- 12,5% от общего числа классов имеют высокие значения метрик связи;
- 4,1% классов с глубокой иерархией;
- в более ранних эмпирических исследованиях [10] предполагается, что классы с большей связью и глубокой иерархией склонны к ошибкам, идентифицированные классы (16,6%) должны быть тщательно проверены во время тестирования;

Таблица 1

Программные метрики для оценки ПС

Название метрики	Расчетная формула	Описание	Экспертное пороговое значение
Количество взвешенных методов на класс (WMC)	$WMC(C_i) = \sum_{i=1}^{NM} L_i$	$L[i]$ – количество строк текста у метода с номером i ; NM – общее количество методов	$NM \geq 20$
Фактор закрытости метода (МНФ)	$MNF = \frac{\sum_{i=1}^{TC} M_h(C_i)}{\sum_{i=1}^{TC} M_d(C_i)}$	$M_h(C_i)$ – количество скрытых методов в классе C_i ; $M_d(C_i)$ – общее количество методов, определенных в классе (унаследованные методы не учитываются)	$MNF \geq 0,5$
Фактор закрытости атрибута (АНФ)	$ANF = \frac{\sum_{i=1}^{TC} A_h(C_i)}{\sum_{i=1}^{TC} A_d(C_i)}$	$A_h(C_i)$ – количество скрытых атрибутов в классе C_i ; $A_d(C_i)$ – общее количество атрибутов, определенных в классе	$ANF \geq 0,9$
Фактор наследования метода (МІФ)	$MIF = \frac{\sum_{i=1}^{TC} M_i(C_i)}{\sum_{i=1}^{TC} M_d(C_i)}$	$M_i(C_i)$ – количество унаследованных и не переопределенных методов в классе C_i ; $M_d(C_i)$ – количество методов, определенных в классе C_i	$0,4 \leq MIF \leq 0,8$
Фактор наследования атрибута (АІФ)	$AIF = \frac{\sum_{i=1}^{TC} A_i(C_i)}{\sum_{i=1}^{TC} A_d(C_i)}$	$A_i(C_i)$ – количество унаследованных атрибутов в классе C_i ; $A_d(C_i)$ – количество атрибутов, определенных в классе C_i	$0,4 \leq AIF \leq 0,7$

Окончание табл. 1

Название метрики	Расчетная формула	Описание	Экспертное пороговое значение
Фактор полиморфизма (POF)	$POF = \frac{\sum_{i=1}^{TC} M_o(C_i)}{\sum_{i=1}^{TC} (M_n(C_i) \cdot DC(C_i))}$	$M_o(C_i)$ – количество унаследованных и переопределенных методов в классе C_i ; $M_n(C_i)$ – количество унаследованных методов, определенных в классе C_i ; $DC(C_i)$ – количество потомков у класса C_i	POF ≤ 0,1
Фактор сцепления (COF)	$COF = \frac{\sum_{i=1}^{TC} \left[\sum_{j=1}^{TC} is_client(C_i, C_j) \right]}{TC^2 - TC}$	$is_clients(C_i, C_j)$ – оценка сцепления классов C_i и C_j ; TC – количество классов в ПС	COF ≤ 0.1
Связанность между классами объектов (CBO)	$CBO = \sum_{i=1}^{TC} \left[\sum_{j=1}^{TC} is_use(C_i, C_j) \right]$	$is_use(C_i, C_j)$ – оценка связанности классов C_i и C_j (все виды отношений – наследование, использование, ассоциация); TC – количество классов в ПС	CBO ≤ 10

Программная реализация алгоритма решения задачи

Схема алгоритма программного модуля расчета метрик качества ПССН АСУ приведена на рисунке 2.

На первом шаге выбирается каталог, в котором находится исследуемый проект и код ПССН. На втором шаге осуществляется выбор групп метрик, которые будут использоваться для оценки качества ПССН. На третьем этапе программа производит загрузку модулей и библиотек для оценки метрик, выбранных пользователем. На четвертом осуществляется расчет метрик для классов и пакетов соответственно. Блок анализа предназначен для расчета комплексных показателей качества (ПК) программных средств и формирования рекомендаций по улучшению качества программного обеспечения в случае выхода значений показателей за пределы допустимых диапазонов. Данный вычислительный блок включает в себя оценку следующих показателей качества: инкапсуляция, наследование, полиморфизм.

Если пользователю необходимо визуализировать полученные данные, программный компонент позволяет отобразить результаты расчетов. Для итерационной оценки качества разрабатываемого ПС формируется архивная копия расчетов с записью в файловую базу данных (БД).

На основе представленного алгоритма была разработана программа. Проверка ее работоспособности, корректности получаемых с ее помощью результатов была осуществлена посредством анализа программного комплекса «Радиус» [8]. Полученные результаты представлены в таблице 2.

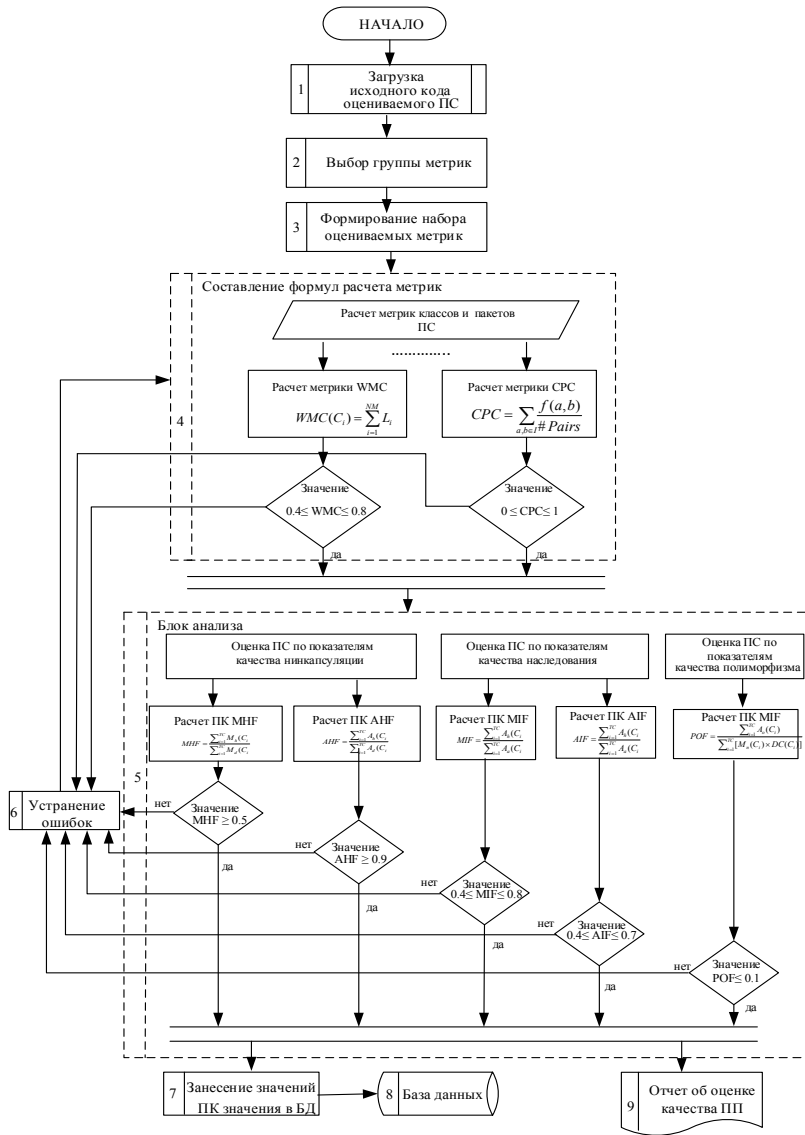


Рис. 2. Алгоритм программного модуля расчета метрик ПССН АСУ

Таблица 2

Результаты анализа ПК «Радиус»

Название класса	Названия и расчетные значения метрик								
	WMC	MHF	ANF	MIF	AIF	POF	COF	MVG	СВО
ConvAzimuthThread	3	1,0	1,0	0,7	0,6	0,05	0,05	42	2
ConvDistColumnThread	3	1,0	1,0	0,7	0,6	0,05	0,05	21	3
Histogram	3	1,0	1,0	0,2	0,3	0,05	0,05	2	3
CalcOpor	3	1,0	1,0	0,2	0,2	0,1	0,05	10	0
RadarImageProc	14	1,0	1,0	0,6	0,5	0,05	0,1	9	7

Киреев А.П., Шаров С.А. Программный компонент расчета показателей...

Оценка метрик для ПС показала малые значения метрики «фактор наследования» ($0,4 \leq \text{MIF} \leq 0,8$) для классов Histogram и CalcOrog, что потребует произвести рефакторинг существующего программного решения. Однако любой рефакторинг (без применения средств автоматизации) скорее всего приведет к возникновению ошибок, и эти ошибки сложно будет обнаружить [10].

Заключение

Проведенный анализ показал, что применение рассмотренного в данной статье программного компонента при разработке ПССН обеспечивает:

- оценку качества программных компонентов на различных этапах жизненного цикла;
- более качественную реализацию программных компонентов с учетом требований ООЯП;
- снижение рисков при сопровождении и эксплуатации ПССН;
- возможности для создания качественных методик проектирования ПС.

Литература

1. Борисов А.А., Краснов С.А., Нечай А.А. Технология блокчейн и проблемы ее применения в различных информационных системах // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». 2018. Вып. 2. С. 63–67.
2. ГОСТ Р ИСО/МЭК 25040–2014. Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQaRE). М.: Стандартинформ, 2014. 34 с.
3. Нечай А.А. Формирование безопасной информационной среды // Актуальные проблемы современности: наука и общество. 2019. № 4 (25). С. 43–44.
4. Нечай А.А., Котиков П.Е. Методика повышения надежности функционирования систем, организованных на перепрограммируемых элементах // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». 2016. Вып. 1-2. С. 87–89.
5. Новиков А.Н., Нечай А.А., Малахов А.В. Математическая модель обоснования вариантов реконфигурации распределенной автоматизированной контрольно-измерительной системы // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». 2016. Вып. 1-2. С. 56–59.
6. Орлов С.А. Технологии разработки программного обеспечения. Разработка сложных программных систем: учебник. СПб.: Питер, 2002. 464 с.
7. Романов В.Ю. Анализ объектно ориентированных метрик для проектирования архитектуры программного обеспечения // International Journal of Open Information Technologies. 2014. Vol. 2, no. 3.
8. Русинов Ю.Л., Титов К.И., Шаров С.А. Программный комплекс обработки и синтеза радиолокационного изображения // Сборник алгоритмов и программ прикладных задач ВКА имени А.Ф. Можайского. СПб., 2013.
9. Шаймарданов А.М., Нечай А.А., Лепехин С.В. Математические модели систем автоматического управления с широтно-импульсной модуляцией // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». 2019. Вып. 2. С. 27–39.
10. Abreu F.B., Melo V. Assessing the Impact of Object-Oriented Design on Software Quality. Proc. 3rd International Symposium on Software Metrics. Berlin, 1996.

11. Aggarwal K.K., Yogesh Singh, Arvinder Kaur, Ruchika Malhotra. Empirical Study of Object-Oriented Metrics // Journal of Object Technology. 2006. Vol. 5, no. 8.
12. Briand L., Daly W., Wust J. Exploring the Relationships Between Design Measures and Software Quality // Journal of Systems and Software. 2000.
13. Chidamber S.R., Kemerer C.F. A Metric Suite for OO Design // IEEE Transactions on Software Engineering, 1994.
14. ISO/IEC 25000:2014 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – Guide to SquaRE: SO/IEC JTC 1 / SC 7 Software and Systems Engineering, 2014.
15. Lorenz M., Kidd J. Object-Oriented Software Metrics: A Practical Guide. Prentice-Hall, 1994.
16. Mishra D. New Inheritance Complexity Metrics for Object-Oriented Software Systems: An Evaluation with Weyuker's Properties // Computing and Informatics. 2011. Vol. 30.
17. Object-Oriented Software Engineering: Measuring and Controlling the Development Process Fernando Brito e Abreu (INESC/ISEG) Rogério Carapuça (INESC/IST). URL: <https://www.researchgate.net/publication/2253619> (date of the application: 11.09.2020).
18. Ponisio María Laura. Exploiting Client Usage to Manage Program Modularity: PhD thesis. University of Bern, 2006 (to the CPC metric algorithm).

Literatura

1. Borisov A.A., Krasnov S.A., Nechaj A.A. Tekhnologiya blokchejn i problemy ee primeneniya v razlichnyh informacionnyh sistemah // Vestnik Rossijskogo novogo universiteta. Seriya "Slozhnye sistemy: modeli, analiz i upravlenie". 2018. Vyp. 2. S. 63–67.
2. GOST R ISO/MEK 25040–2014. Informacionnye tekhnologii. Sistemnaya i programm-naya inzheneriya. Trebovaniya i ocenka kachestva sistem i programmnoho obespecheniya (SQuaRE). M.: Standartinform, 2014. 34 s.
3. Nechaj A.A. Formirovanie bezopasnoj informacionnoj sredy // Aktual'nye problemy sovremennosti: nauka i obshchestvo. 2019. № 4 (25). S. 43–44.
4. Nechaj A.A., Kotikov P.E. Metodika povysheniya nadezhnosti funkcionirovaniya sistem, organizovannyh na pereprogrammiruemyh elementah // Vestnik Rossijskogo novogo universiteta. Seriya "Slozhnye sistemy: modeli, analiz i upravlenie". 2016. Vyp. 1-2. S. 87–89.
5. Novikov A.N., Nechaj A.A., Malahov A.V. Matematicheskaya model' obosnovaniya variantov rekonfiguracii raspredelennoj avtomatizirovannoj kontrol'no-izmeritel'noj sistemy // Vestnik Rossijskogo novogo universiteta. Seriya "Slozhnye sistemy: modeli, analiz i upravlenie". 2016. Vyp. 1-2. S. 56–59.
6. Orlov S.A. Tekhnologii razrabotki programmnoho obespecheniya. Razrabotka slozhnyh programmnyh sistem: uchebnyk. SPb.: Piter, 2002. 464 s.
7. Romanov V.Yu. Analiz ob"ektno orientirovannyh metrik dlya proektirovaniya arhitektury programmnoho obespecheniya // International Journal of Open Information Technologies. 2014. Vol. 2, no. 3.
8. Rusinov Yu.L., Titov K.I., Sharov S.A. Programmnyj kompleks obrabotki i sinteza radio-lokacionnogo izobrazheniya // Sbornik algoritmov i programm prikladnyh zadach VKA imeni A.F. Mozhajsogo. SPb., 2013.
9. Shajmardanov A.M., Nechaj A.A., Lepekhin S.V. Matematicheskie modeli sistem avtomaticheskogo upravleniya s shirotno-impul'snoj modulyaciej // Vestnik Rossijskogo novogo universiteta. Seriya "Slozhnye sistemy: modeli, analiz i upravlenie". 2019. Vyp. 2. S. 27–39.
10. Abreu F.B., Melo V. Assessing the Impact of Object-Oriented Design on Software Quality. Proc. 3rd International Symposium on Software Metrics. Berlin, 1996.

Набиев С.Р., Набиев Б.Р. Система верификации удостоверения личности...

11. Aggarwal K.K., Yogesh Singh, Arvinder Kaur, Ruchika Malhotra. Empirical Study of Object-Oriented Metrics // Journal of Object Technology. 2006. Vol. 5, no. 8.
12. Briand L., Daly W., Wust J. Exploring the Relationships Between Design Measures and Software Quality // Journal of Systems and Software. 2000.
13. Chidamber S.R., Kemerer C.F. A Metric Suite for OO Design // IEEE Transactions on Software Engineering, 1994.
14. ISO/IEC 25000:2014 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – Guide to SquaRE: SO/IEC JTC 1 / SC 7 Software and Systems Engineering, 2014.
15. Lorenz M., Kidd J. Object-Oriented Software Metrics: A Practical Guide. Prentice-Hall, 1994.
16. Mishra D. New Inheritance Complexity Metrics for Object-Oriented Software Systems: An Evaluation with Weyuker's Properties // Computing and Informatics. 2011. Vol. 30.
17. Object-Oriented Software Engineering: Measuring and Controlling the Development Process Fernando Brito e Abreu (INESC/ISEG) Rogério Carapuça (INESC/IST). URL: <https://www.researchgate.net/publication/2253619> (date of the application: 11.09.2020).
18. Ponisio María Laura. Exploiting Client Usage to Manage Program Modularity: PhD thesis. University of Bern, 2006 (to the CPC metric algorithm).

DOI: 10.25586/RNUV9187.20.04.P.065

УДК 004.4

С.Р. Набиев, Б.Р. Набиев

СИСТЕМА ВЕРИФИКАЦИИ УДОСТОВЕРЕНИЯ ЛИЧНОСТИ НА ОСНОВЕ ТЕХНОЛОГИИ BLOCKCHAIN

Посвящено вопросу обеспечения безопасности персональных данных с помощью системы верификации, основанной на технологии распределенного регистра. Актуальность соответствующей проблемы обусловлена в первую очередь ростом числа преступлений, связанных с подделкой документов и соответствующей задачей обеспечения безопасности персональных данных, регламентированной рядом основополагающих нормативных правовых актов. Уделяется значительное внимание техническим вопросам созданной системы верификации. Рассмотрен алгоритм, лежащий в основе разработанных приложений для проверки данных с применением технологии распределенного регистра.

Ключевые слова: блокчейн, верификация данных, кибербезопасность, QR-код, хеш-функция.

S.R. Nabiev, B.R. Nabiev

ID VERIFICATION SYSTEM BASED ON BLOCKCHAIN TECHNOLOGY

The purpose of this paper is to go into detail on how to ensure the security of personal data based on a verification system based on distributed registry technology. The relevance of the corresponding problem is primarily due to the increase in the number of crimes related to forgery of documents and the corresponding task of ensuring the security of personal data regulated by several fundamental regulatory legal acts. For example, GDPR provides residents of the European Union with the potential of complete