

ФОРМИРОВАНИЕ ОПТИМАЛЬНОГО КОМПЛЕКТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЯХ

А.В. Бобровских¹, Ю.В. Бондаренко²

¹ Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия имени профессора Н.Е. Жуковского и Ю.А. Гагарина», г. Воронеж, Россия,

² Воронежский государственный университет, г. Воронеж, Россия

Рассмотрена проблема определения номенклатуры оптимального комплекта прикладного программного обеспечения (ПО), необходимого для реализации образовательных программ в высших учебных заведениях (вуз), путем количественной оценки его качества. **Цель исследования:** разработка механизма комплексной оценки качества ПО в рамках выбранного класса, позволяющего оказать поддержку в принятии обоснованного решения при выборе типового комплекта ПО вуза. **Материалы и методы.** Предлагаемый механизм основывается на использовании методов оптимизации и экспертных оценках. На первом этапе механизма уточняются базовые требования, учитывающие специфику работы образовательной организации. Формируется набор программ (альтернатив) для выбранного класса ПО с последующим определением на базе экспертной информации характерных для него ключевых функций. Далее по предложенным в работе показателям качества ПО осуществляется их оценка для каждой альтернативы. Показатели качества ПО представляют собой иерархическую трехуровневую систему (критерий, метрика, оценочный элемент), в которой показатели вышестоящих уровней определяются через показатели нижестоящих уровней. Интегральная оценка качества альтернативы определяется как сумма оценок ее критериев качества. На завершающем этапе решается задача целочисленного линейного программирования по определению оптимальной номенклатуры альтернатив в рамках соответствующего класса ПО, исходя их условий максимизации качества альтернативы, содержащей в себе требуемый набор функций, и минимизации их количества. **Результаты.** Разработан механизм определения оптимальных альтернатив для формирования типового набора ПО с целью реализации необходимых образовательных программ вуза. Формализованы основные показатели качества ПО и предложен математический аппарат по их оценке. **Заключение.** Предложенный механизм позволяет осуществить количественную оценку качества ПО в рамках его класса с последующим выбором наиболее рациональных для вуза программных средств. Такая оценка позволит упростить задачу по принятию обоснованного решения при формировании типового комплекта ПО, необходимого для реализации требований федеральных образовательных стандартов, и в целом повысит эффективность системы координации информатизации по данному направлению.

Ключевые слова: программное обеспечение, альтернатива, показатели качества, оценка качества, типовой комплект, информатизация вуза.

Введение

Согласно типовым требованиям к материально-техническому и учебно-методическому обеспечению, определенным федеральными государственными образовательными стандартами высшего образования, организация должна быть обеспечена в том числе и необходимым комплектом лицензионного и свободно распространяемого ПО. Состав такого комплекта определяется в рабочих программах дисциплин (модулей) и подлежит обновлению при необходимости. Таким образом, от данного компонента зависит качество и эффективность подготовки обучающихся.

На текущий момент механизм формирования оптимального комплекта ПО в вузах явно не описан и слабо структурирован. При этом определен ряд ограничений при осуществлении его

закупки и установки на объекты информатизации указанных организаций [1–3], основными из которых являются:

- запрет на допуск ПО, происходящего из иностранных государств;
- дефицит денежных средств на соответствующую статью расходов;
- ограничения по допуску ПО на объекты информатизации, предназначенных для обработки информации ограниченного доступа.

Также следует учитывать, что Российская Федерация (РФ) находится под определенным политическим давлением, выражающимся, в частности, в ряде санкций [4], затрагивающих ИТ-сферу. Результатом таких мер является ограничение (блокировка) для ряда организаций РФ (в том числе научных и образовательных) возможности приобретения программных продуктов от соответствующих иностранных разработчиков.

Учитывая сложившуюся ситуацию, перед вузами стоит задача без потерь к требуемому уровню подготовки обучающихся осуществить отбор наиболее перспективных (рациональных) программных продуктов, в меньшей степени подверженных указанным ограничениям. Для решения данной задачи необходимо разработать механизм комплексной оценки качества ПО в рамках выбранного класса. Такая оценка позволит формализовать процесс выбора выигрышных программных решений среди многообразия альтернатив и упростить задачу по принятию обоснованного решения при выборе типового комплекта ПО вуза.

Описание подхода

Предлагаемый механизм основывается на математических методах оптимизации [5–9], общих положениях по оценке качества программных средств, установленных стандартами различного статуса [10–12], а также экспертных оценках.

Будем предполагать, что для некоторого вуза определена потребность в ПО, выраженная в конечном наборе его классов m . Множество классов обозначим через $K = \{k_q\}$, где $q = 1, \dots, m$ – порядковый номер класса. В рамках данной статьи под классом ПО будет пониматься группа, которая объединяет ПО, имеющее аналогичные функциональные характеристики (возможности). Номенклатура классов устанавливается в соответствии с классификатором программ для электронных вычислительных машин и баз данных, утвержденным приказом Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации 22.11.2020 г. № 486. Так, например, среди офисных приложений и прикладного ПО популярными (востребованными) являются классы «Редакторы презентаций», «Табличные редакторы», «Текстовые редакторы», «Мультимедийное программное обеспечение», «Информационные системы для решения специфических отраслевых задач», «Справочно-правовые системы».

Пусть также для каждого класса k_q сформирован набор программ p_q (альтернатив), из которых необходимо выбрать оптимальные для данного класса. Множество альтернатив в рамках k_q -класса обозначим через $A_q = \{a_g^q\}$, где $g = 1, \dots, p_q$ – порядковый номер альтернативы.

Известен базовый перечень требований организации, которым должны соответствовать указанные альтернативы, количество которых равно b . Номенклатуру основных требований организации обозначим через $T = \{t_y\}$, где $y = 1, \dots, b$ – порядковый номер требования (ограничения). Из множества A_q отбираются только альтернативы t_q , соответствующие базовым требованиям. Множество таких альтернатив обозначим через $A_q^T = \{a_j^q\}$, где $j = 1, \dots, t_q$ – порядковый номер альтернативы, соответствующей базовым требованиям организации, а их комплексную оценку (вес) – через W_j^q .

Будем полагать, что на основе экспертной информации и соответствующей нормативной базы для k_q класса может быть разработана матрица типовых функций программы $F_q = (f_{ij}^q)$ (матрица классификационных признаков), где $i = 1, \dots, n_q$ – порядковый номер функции, характеризующей класс k_q , $j = 1, \dots, t_q$.

Каждая альтернатива a_j^q может либо содержать в себе необходимый функционал, либо нет. Если альтернатива a_j^q позволяет реализовать f_i^q функцию k_q -класса, то элемент матрицы $f_{ij}^q = 1$, если нет, то $f_{ij}^q = 0$.

Требуется сформировать механизм комплексной оценки альтернатив a_j^q в рамках выбранного класса для упрощения задачи по принятию обоснованного решения при выборе типового комплекта ПО вуза. Данную задачу можно записать следующим образом:

$$\sum_{j=1}^{t_q} x_j^q \rightarrow \min, \quad (1)$$

где $x_j^q = \begin{cases} 1, & \text{если } a_j^q \text{ альтернатива включается в } k_q\text{-класс;} \\ 0, & \text{иначе.} \end{cases}$

$$\sum_{j=1}^{t_q} W_j^q x_j^q \rightarrow \max, \quad (2)$$

где W_j^q – оценка (вес) a_j^q альтернативы.

$$\sum_{j=1}^{t_q} f_{ij}^q x_j^q \geq 1, \quad i = 1, \dots, n_q, \quad (3)$$

где $f_{ij}^q = \begin{cases} 1, & \text{если } a_j^q \text{ альтернатива реализует функцию } f_i^q \text{ для } k_q\text{-класса;} \\ 0, & \text{иначе.} \end{cases}$

Переменными задачи дискретной оптимизации (1)–(3) является x_j^q .

Решение данной задачи предлагается осуществить в 4 этапа.

Этап 1. Проверка соответствия сформированного набора альтернатив базовым (входным) требованиям организации.

Начальным этапом выбора лучшей альтернативы является их проверка на соответствие базовым требованиям организации. Перечень типовых требований (ограничений), которые в большинстве случаев необходимо учитывать при снабжении вузов ПО, а также их оценка представлены в табл. 1. Если для данной организации часть приведенных в табл. 1 требований избыточны, то они не учитываются (в расчете не участвуют).

Таблица 1

Базовые требования по допуску программного обеспечения на объекты информатизации организации

Table 1

Basic requirements for the admission of software to the objects of informatization of the organization

№ п/п	Перечень требований	Оценка
1	Соответствие требованиям образовательного стандарта, t_1	$w_{1g}^T = \begin{cases} 1, & \text{если программа соответствует требованиям;} \\ 0, & \text{в противном случае.} \end{cases}$ где w_{1g}^T – оценка соответствия g -й программы t_1 требованию организации
2	Наличие допуска ПО на ОИ, обрабатывающих сведения, составляющие государственную тайну, t_2	$w_{2g}^T = \begin{cases} 1, & \text{если программа имеет соответствующий допуск;} \\ 0, & \text{в противном случае.} \end{cases}$ где w_{2g}^T – оценка соответствия g -й программы t_2 требованию организации

№ п/п	Перечень требований	Оценка
3	Выполнение запрета на допуск ПО, происходящего из иностранных государств, t_3	$w_{3g}^T = \begin{cases} 1, & \text{если программа включена в ЕРРП или свободная;} \\ 0, & \text{в противном случае.} \end{cases}$ где w_{3g}^T – оценка соответствия g -й программы t_4 требованию организации
4	Аппаратная совместимость, t_4	$w_{4g}^T = \begin{cases} 1, & \text{если совместима с аппаратными средствами;} \\ 0, & \text{в противном случае.} \end{cases}$ где w_{4g}^T – оценка соответствия g -й программы t_5 требованию организации
5	Совместимость с ОС, t_5	$w_{5g}^T = \begin{cases} 1, & \text{если программа совместима с ОС организации;} \\ 0, & \text{в противном случае.} \end{cases}$ где w_{5g}^T – оценка соответствия g -й программы t_6 требованию организации
6	Требование для ПО наличия доступа к сети «Интернет», t_6	$w_{6g}^T = \begin{cases} 1, & \text{если способ активации и использования ПО может} \\ & \text{быть реализован пользователем;} \\ 0, & \text{в противном случае.} \end{cases}$ где w_{6g}^T – оценка соответствия g -й программы t_7 требованию организации

Итоговую оценку соответствия a_g^q -программы требованиям организации t_y обозначим через W_g^T :

$$W_g^T = \prod_{y=1}^b w_{yg}^T, \tag{4}$$

где w_{yg}^T – оценка соответствия g -й программы y -му требованию организации.

Из множества A_q отбираются только те альтернативы t_q , для которых $W_g^T \neq 0$, т. е. соответствующие базовым требованиям. Таким образом, получаем новое множество альтернатив A_q^T , такое что $A_q^T \subseteq A_q$.

Этап 2. Определение значений показателей качества альтернатив.

Оценка оптимальности альтернатив осуществляется посредством определения значений показателей качества для них. Показатели качества альтернатив представляют собой иерархическую трехуровневую систему, в которой показатели вышестоящих уровней определяются через показатели нижестоящих уровней.

Первый (верхний) уровень представляет собой *критерии качества ПО*: функциональные возможности (F_q), надежность (R), адаптируемость (C), условия использования и распространения (лицензия) (L), удобство использования (U), статус проекта (H). Критерии качества программного обеспечения по степени важности принимаются равнозначными.

Критерии качества определяют соответствующими *метриками* (второй уровень): функции (f_1, \dots, f_n), устойчивость к ошибке (r_1), возможность повторного старта с точки останова (r_2), степень аппаратной совместимости (c_1), степень программной совместимости (c_2), модель дохода

(стоимость программы) (l_1), срок использования программы (l_2), количество рабочих мест (l_3), техническая поддержка (l_4), рейтинг программы в тематических обзорах (u_1), документация для освоения (u_2), качество помощи (легкость освоения) (u_3), локализация программ, происходящих из иностранных государств (u_4), наличие опыта работы у преподавательского состава с программой (u_5), совместимость программы с аналогами (u_6), состояние проекта на текущий год (h_1), история актуального проекта (h_2). Метрики характеризуются двумя числовыми параметрами – количественным значением (оценкой) w_{ij}^Ψ и весовым коэффициентом s_i^Ψ , где $i = 1, \dots, n_\Psi$ – порядковый номер метрики, $j = 1, \dots, t_q$ – порядковый номер альтернативы, $\Psi = \{F_q, R, C, L, U, H\}$ – признак критерия качества, n_Ψ – количество метрик, содержащихся в критерии Ψ .

Важность (вес) метрики, уточняется экспертной группой по средствам весового коэффициента s_i^Ψ и зависит от специфики конкретной организации. Предлагается в качестве значений коэффициента s_i^Ψ использовать следующие:

$$s_i^\Psi = \begin{cases} 1, & \text{если } i\text{-я метрика для программы является основной (О);} \\ 0,5, & \text{если } i\text{-я метрика для программы является второстепенной (В);} \\ 0, & \text{в противном случае (} i\text{-я метрика является избыточной (И)).} \end{cases} \quad (5)$$

Метрики состояются из оценочных элементов (третий уровень): процессор (\dot{c}_1), оперативная память (\dot{c}_2), память видеокарты (\dot{c}_3), память на диске (\dot{c}_4), наличие пользовательской документации для освоения (\dot{u}_1), полнота описания программы в пользовательской документации (\dot{u}_2), точность пользовательской документации для освоения (\dot{u}_3), возможность освоения программы на контрольном примере при помощи ЭВМ (\dot{u}_4), наличие в программе встроенной справки (функция HELP) (\dot{u}_5), наличие в открытом доступе других методических материалов (\dot{u}_6). Оценочный элемент определяет заданное в метрике свойство. Для метрики, определенной одним оценочным элементом, третий уровень опускается.

Для показателей качества на всех уровнях (критерии, метрики, оценочные элементы) принимается единичная шкала оценки от 0 до 1.

При оценке критериев качества применяется традиционный подход, базирующийся на аддитивном методе, когда интегральная оценка критерия качества W_j^Ψ альтернативы a_j^q будет определяться как сумма частных оценок (оценок метрик w_{ij}^Ψ), умноженных на веса (коэффициент s_i^Ψ):

$$W_j^\Psi = \sum_{i=1}^{n_\Psi} s_i^\Psi w_{ij}^\Psi. \quad (6)$$

Нормализация данной оценки осуществляется путем ее деления на количество метрик n_Ψ , содержащихся в критерии Ψ :

$$\tilde{W}_j^\Psi = W_j^\Psi / n_\Psi. \quad (7)$$

Процесс оценки метрик зависит от их наполнения. Если метрика формируется из нескольких оценочных элементов o_M , то результат ее оценки w_{ij}^Ψ определяется по формуле

$$w_{ij}^\Psi = \sum_{d=1}^{o_M} w_{dj}^M / o_M, \quad (8)$$

где w_{dj}^M – оценка d -го оценочного элемента для j -й альтернативы метрики M ;

$$M = \{f_1, \dots, f_n, r_1, r_2, c_1, c_2, l_1, \dots, l_4, u_1, \dots, u_6, h_1, h_2\} \text{ – признак метрики.}$$

Если метрика определена одним оценочным элементом (третий уровень отсутствует), то для ее оценки применяется преимущественно экспертная информация или выражения:

$$w_{ij}^{\Psi} = \lambda_i \frac{\theta_{ij} - \min_j(\theta_{ij})}{\max_j(\theta_{ij}) - \min_j(\theta_{ij})} \text{ -- для метрик с прямой шкалой;} \quad (9)$$

$$w_{ij}^{\Psi} = \lambda_i \frac{\max_j(\theta_{ij}) - \theta_{ij}}{\max_j(\theta_{ij}) - \min_j(\theta_{ij})} \text{ -- для метрик с обратной шкалой,} \quad (10)$$

где θ_{ij} – числовое значение i -й метрики для j -й программы;

$\lambda_i \in \{0, 5; 1\}$ – корректирующий коэффициент.

Оценка третьего уровня (оценочные элементы) осуществляется на основе информации, представленной преимущественно разработчиками ПО на соответствующих сайтах в сети «Интернет», экспертной информации, а также выражений (9), (10).

Этап 3. Определение комплексной оценки качества альтернатив.

На 3-м этапе определяется комплексная оценка W_j^q для каждой альтернативы из множества A_q^T в рамках класса k_q :

$$W_j^q = \tilde{W}_j^F + \tilde{W}_j^R + \tilde{W}_j^C + \tilde{W}_j^L + \tilde{W}_j^U + \tilde{W}_j^H. \quad (11)$$

Этап 4. Определение оптимальной номенклатуры альтернатив для требуемого класса ПО.

На завершающем этапе алгоритма методом Гомори [4, 5] определяется минимальный комплект соответствующих альтернатив, имеющих максимальную оценку W_j^q и содержащих в себе все необходимые функции (статус метрики определен как «основная»). Особенности реализации алгоритма заключаются в следующем. Предполагаем, что среди множества альтернатив A_q^T есть альтернативы, содержащие каждая в себе полный набор основных функций, из которых требуется выбрать одну с максимальной оценкой W_j^q . Тогда условие для поиска данного решения будет иметь вид:

$$\begin{cases} \sum_{j=1}^{t_q} W_j^q x_j^q \rightarrow \max, \\ \sum_{j=1}^{t_q} f_{ij}^q x_j^q \geq 1; \\ \sum_{j=1}^{t_q} x_j^q = 1, \end{cases} \quad (12)$$

где $i = 1, \dots, n_q^o$ – порядковый номер функции, определенной как «основная».

Если для системы (12) существует решение, то оно будет определять оптимальную альтернативу для данного класса и соответственно включаться в типовой комплект ПО организации.

Если система (12) не имеет решений (среди множества альтернатив отсутствует программа с требуемым набором функций), то максимально допустимое количество программ в классе увеличивается до двух:

$$\begin{cases} \sum_{j=1}^{t_q} f_{ij}^q x_j^q \geq 1; \\ \left(\sum_{j=1}^{t_q} x_j^q \leq 2 \right) \wedge (x_j^q \leq 1), \end{cases} \quad (13)$$

а при отсутствии положительного результата до трех:

$$\left\{ \begin{array}{l} \sum_{j=1}^{t_q} f_{ij}^q x_j^q \geq 1; \\ \left(\sum_{j=1}^{t_q} x_j^q \leq 3 \right) \wedge (x_j^q \leq 1). \end{array} \right. \quad (14)$$

Если при ограничении в три программы для класса решение не может быть найдено, то необходимо либо пересмотреть (расширить) соответствующую исходную базу альтернатив с большим набором функциональных возможностей, либо сократить число ключевых (основных) функций (перевести их в статус «второстепенные»), характеризующих данный класс.

Дальнейшее увеличение количества программ в классе, по мнению авторов статьи, нецелесообразно по следующим причинам:

1. Увеличение периода адаптации сотрудников организации к новому ПО. Каждая новая программа (интерфейс) будет требовать от пользователя дополнительного времени на получение опыта работы в ней, что вызовет снижение производительности труда на данный период.

2. Проблемы администрирования. Поддержка большого количества программ увеличит нагрузку на советующие ИТ-службы организации (системных администраторов).

3. Проблемы совместимости форматов файлов. Различие в форматах создаст дополнительные сложности стыковки ПО от разных разработчиков, что приведет к увеличению затрат на совместимость результатов работы, усложнит процесс унификации ПО и в целом управление информационно-телекоммуникационной инфраструктурой вуза.

4. Увеличение времени и средств на разработку методического материала по работе с ПО.

Если системы (13) или (14) имеют решение, то оно будет определять соответственно две или три выигранных (оптимальных) альтернативы для данного класса. Отобранные таким образом для каждого класса альтернативы формируют оптимальный комплект программ организации.

Практические расчеты

Рассмотрим работу механизма определения оптимальной номенклатуры (комплекта) ПО на примере одного из вузов Министерства обороны Российской Федерации.

Пусть согласно требованиям к условиям реализации программы специалитета организацию необходимо обеспечить прикладным программным обеспечением, соответствующим классу «Редакторы презентаций». В рамках данного класса определены десять программ a_1, \dots, a_{10} , из которых необходимо выбрать оптимальные и для которых известны функциональные и эксплуатационные характеристики.

На средства вычислительной техники объектов информатизация вуза установлена операционная система (ОС) Astra Linux. Характеристики данных средств определены следующими параметрами (системные требования): частота процессора – 2,2 ГГц, объем оперативной памяти – 4 Гб, память видеокарты – 2 Гб, память на диске – 350 Гб. Обработка сведений, составляющих государственную тайну, на данных средствах не ведется. Приобретение ПО осуществляется в соответствии с постановлением Правительства РФ от 16 ноября 2015 г. № 1236.

Таким образом, обозначены базовые требования организации, в частности, по выполнению запрета на допуск ПО, происходящего из иностранных государств, совместимости ПО с ОС и аппаратной частью организации.

Этап 1. На начальном этапе осуществляется проверка на соответствие программы a_g , указанным выше требованиям. Если для программы a_g не выполняется одно из требований, то она в расчет не берется (исключается из потенциального комплекта ПО).

Пусть альтернатива a_1 включена в единый реестр российских программ [13] ($w_{41}^T = 1$) и может работать под управлением ОС семейства Windows, Linux и Mac ($w_{61}^T = 1$), на средствах вычислительной техники с частотой процессора 1 ГГц, оперативной памятью 2 Гб, видеокарты в 512 Мб и занимать место на жестком диске порядка 3 Гб ($w_{51}^T = 1$). Тогда согласно выражению (4) оценка соответствия альтернативы a_1 требованиям организации $W_1^T = 1$. Следовательно, программа полностью соответствует базовым требованиям вуза и может быть использована на его объектах информатизации.

Информатика и вычислительная техника

Альтернатива a_2 включена в единый реестр российских программ ($w_{42}^T = 1$) и может работать под управлением ОС семейства Windows ($w_{62}^T = 0$), на средствах вычислительной техники с частотой процессора 1 ГГц, оперативной памятью 1,5 Гб, видеокартой в 512 Мб и занимать место на жестком диске порядка 2 Гб ($w_{52}^T = 1$). Тогда согласно выражению (4) соответствующая оценка $W_2^T = 0$ (программа не совместима с ОС организации) и, следовательно, альтернатива a_2 исключается из дальнейшего рассмотрения.

Аналогично осуществляется проверка оставшихся альтернатив. Пусть в ходе проверки соответствия были отобраны семь альтернатив.

Этап 2. На данном этапе определяются значения показателей качества для отобранных на первом этапе альтернатив. В качестве примера рассмотрим расчет оценок для трех критериев качества: «Функциональные возможности», «Адаптируемость», «Условия использования и распространения (лицензия)». Остальные показатели качества для данной организации будем считать избыточными ($s_i^{\Psi} = 0$).

В соответствии с постановлением Правительства РФ от 23.03.2017 г. № 325 ключевыми (основными) функциональными характеристиками для класса «Редактор презентаций» являются: поддержка форматов Open Document Format (f_1), Office Open XML (OOXML, PPTX) (f_2), выбор различных шрифтов (f_3), выравнивание текста (по центру, ширине, левой и правой границе зоны размещения информации) (f_4), вставка и удаление из слайда графических элементов, аудио- и видеоинформации (f_5), настройка демонстрации слайдов (f_6). В качестве второстепенных функций для данного класса, учитывая экспертную информацию и специфику организации, определим поддержку формата PDF (функция f_7) и использование внешних электронных орфографических словарей русского языка (f_8). Для демонстрации работы механизма указанный набор классификационных признаков (f_1, \dots, f_8) будем считать достаточным.

Функции f_1, \dots, f_8 являются метриками для критерия качества «Функциональные возможности, F » и оцениваются по следующему правилу:

$$w_{ij}^F = \begin{cases} 1, & \text{если программа реализует требуемую функцию;} \\ 0, & \text{иначе,} \end{cases}$$

где w_{ij}^F – оценка i -й метрики для j -й программы критерия F .

Будем полагать, что отобранные альтернативы позволяют реализовать функционал, определенный табл. 2.

Таблица 2

Функциональные возможности альтернатив для класса «Редакторы презентаций»

Table 2

Functionality of alternatives for the “Presentation Editors” class

Метрика	Коеф. s_i^F	Наличие (+)/отсутствие(-) у альтернативы требуемой функции						
		a_1	a_2	a_3	a_4	a_5	a_6	a_7
f_1	О	–	–	+	+	+	–	+
f_2	О	+	+	+	–	–	+	+
f_3	О	+	+	+	–	+	+	+
f_4	О	+	–	+	+	–	+	–
f_5	О	+	+	+	+	+	–	+
f_6	О	–	+	–	+	–	+	–
f_7	В	+	–	+	–	+	–	+
f_8	В	+	+	–	–	–	+	+

Тогда оценка критерия качества F для альтернативы a_j вычисляется согласно выражению (6). Определим указанную оценку для альтернативы a_1 :

$$W_1^F = \sum_{i=1}^8 s_i^F w_{i1}^F = 5.$$

Далее осуществляется нормирование данной оценки в соответствии с выражением (7):

$$\tilde{W}_1^F = W_1^F / n_F = 0,625.$$

Аналогичным образом определяются оценки других альтернатив:

$$\tilde{W}_2^F = 0,563, \tilde{W}_3^F = 0,688, \tilde{W}_4^F = 0,500, \tilde{W}_5^F = 0,438, \tilde{W}_6^F = 0,563, \tilde{W}_7^F = 0,625.$$

Для оценки критерия «Адаптируемость, C » необходимо вычислить значение метрик «Степень аппаратной совместимости, c_1 » и «Степень программной совместимости, c_2 ».

Оценка метрики c_1 зависит от четырех оценочных элементов «Процессор, \dot{c}_1 », «Оперативная память, \dot{c}_2 », «Память видеокарты, \dot{c}_3 », «Память на диске, \dot{c}_4 », расчет которых осуществляется в соответствии с выражением (10). Корректирующий коэффициент $\lambda_i = 1$.

Метрика c_2 определена одним оценочным элементом (третий уровень отсутствует) и зависит от количества дополнительных программ, необходимых для функционирования a_j -альтернативы. Дополнительные программы принимаются равнозначными по условиям их применения и распространения. Если они отличаются по данному показателю, то в оценку включаются только не-свободные программы. Расчет метрики c_2 производится аналогично расчету оценочных элементов метрики c_1 .

Пусть для альтернатив a_1, \dots, a_7 определены системные требования, представленные в табл. 3.

Системные требования

Таблица 3

System requirements

Table 3

Параметр	Характеристика альтернативы						
	a_1	a_2	a_3	a_4	a_5	a_6	a_7
Тактовая частота процессора, ГГц	1	1,2	1	1,4	1,3	0,9	1,2
Объем оперативной памяти, Мб	1024	2048	1536	2048	2048	512	1536
Объем памяти видеокарты, Мб	512	512	512	640	512	256	512
Свободное место на жестком диске, Гб	3	3,5	2	3	2,5	1	2
Число доп. программ, шт.	1	1	0	1	2	0	1

Тогда соответствующие оценки оценочных элементов для альтернативы a_1 равны:

$$w_{11}^{c_1} = \left(\max_j (\dot{c}_{1j}) - \dot{c}_{11} \right) / \left(\max_j (\dot{c}_{1j}) - \min_j (\dot{c}_{1j}) \right) = 0,800;$$

$$w_{21}^{c_1} = \left(\max_j (\dot{c}_{2j}) - \dot{c}_{21} \right) / \left(\max_j (\dot{c}_{2j}) - \min_j (\dot{c}_{2j}) \right) = 0,667;$$

$$w_{31}^{c_1} = \left(\max_j (\dot{c}_{3j}) - \dot{c}_{31} \right) / \left(\max_j (\dot{c}_{3j}) - \min_j (\dot{c}_{3j}) \right) = 0,333;$$

$$w_{41}^{c_1} = \left(\max_j (\dot{c}_{4j}) - \dot{c}_{41} \right) / \left(\max_j (\dot{c}_{4j}) - \min_j (\dot{c}_{4j}) \right) = 0,200.$$

Оценка метрики c_1 для альтернативы a_1 рассчитывается согласно формуле (8):

$$w_{11}^C = \sum_{d=1}^{o_{c_1}} w_{d1}^{c_1} / o_{c_1} = 0,500.$$

Оценка метрики c_2 для альтернативы a_1 рассчитывается по формуле (10):

$$w_{21}^C = \left(\max_j (c_{2j}) - c_{21} \right) / \left(\max_j (c_{2j}) - \min_j (c_{2j}) \right) = 0,500.$$

Пусть метрики c_1 и c_2 определены как основные ($s_i^C = 1$).

Тогда согласно выражению (6) оценка критерия качества C для альтернативы a_1 равна:

$$W_1^C = \sum_{i=1}^2 s_i^C w_{i1}^C = 1,000.$$

Информатика и вычислительная техника

Далее осуществляется нормирование данной оценки в соответствии с выражением (7):

$$\tilde{W}_1^C = W_1^C / n_C = 0,500.$$

Аналогичным образом рассчитываются соответствующие оценки для альтернатив a_2, \dots, a_7 :

$$\tilde{W}_2^C = 0,342, \tilde{W}_3^C = 0,759, \tilde{W}_4^C = 0,275, \tilde{W}_5^C = 0,117, \tilde{W}_6^C = 1,000, \tilde{W}_7^C = 0,459.$$

Критерий «Условия использования и распространения (лицензия), L » оценивается метриками «Модель дохода (стоимость программы), l_1 », «Срок использования программы, l_2 », «Количество рабочих мест, l_3 », «Техническая поддержка (ТП), l_4 ».

Оценка метрики l_1 зависит от стоимости альтернативы a_j :

$$w_{1j}^L = \begin{cases} 0,5 \left(\max_j(l_{1j}) - l_{1j} \right) / \left(\max_j(l_{1j}) - \min_j(l_{1j}) \right), & \text{если программа платная;} \\ 1, & \text{иначе.} \end{cases}$$

Оценка метрики l_2 зависит от срока действия неисключительных прав на использование альтернативы a_j :

$$w_{2j}^L = \begin{cases} 0,5 \left(l_{2j} - \min_j(l_{2j}) \right) / \left(\max_j(l_{2j}) - \min_j(l_{2j}) \right), & \text{если срок ограничен;} \\ 1, & \text{иначе.} \end{cases}$$

Оценка метрики l_3 определяется аналогично метрике l_2 и зависит от числа рабочих мест (РМ), на которых лицензией разрешено использовать альтернативу a_j :

$$w_{3j}^L = \begin{cases} 0,5 \left(l_{3j} - \min_j(l_{3j}) \right) / \left(\max_j(l_{3j}) - \min_j(l_{3j}) \right), & \text{если кол. РМ ограничено;} \\ 1, & \text{иначе.} \end{cases}$$

Метрика l_4 оценивается по следующему принципу:

$$w_{4j}^L = \begin{cases} 1, & \text{если ТП входит в указанную стоимость программы;} \\ 0,5, & \text{если ТП не входит в указанную стоимость;} \\ 0, & \text{если ТП для данной программы не предусмотрена.} \end{cases}$$

Пусть условия лицензий на приобретение и использование альтернатив a_1, \dots, a_7 определены табл. 4.

Таблица 4

Условия лицензии на использование программы

Table 4

Terms of the license to use the program

Параметр	Характеристика альтернативы						
	a_1	a_2	a_3	a_4	a_5	a_6	a_7
Стоимость программы, руб.	10 000	7000	0	1000	3000	0	13 000
Срок использования программы, г.	Без огран.	Без огран.	Без огран.	1	3	Без огран.	5
Количество рабочих мест, шт.	1	1	Без огран.	1	1	Без огран.	5
Статус ТП: + входит в цену ± не входит в цену – отсутствует	+	+	–	±	+	–	+

Тогда соответствующие оценки метрик для альтернативы a_1 равны:

$$w_{11}^L = 0,5 \left(\max_j(l_{1j}) - l_{1j} \right) / \left(\max_j(l_{1j}) - \min_j(l_{1j}) \right) = 0,125; w_{21}^L = 1;$$

$$w_{31}^L = 0,5 \left(l_{31} - \min_j (l_{3j}) \right) / \left(\max_j (l_{3j}) - \min_j (l_{3j}) \right) = 0; \quad w_{41}^L = 1.$$

Пусть метрики l_1 , l_2 и l_3 определены как основные ($s_1^L = 1$, $s_2^L = 1$, $s_3^L = 1$), а метрика l_4 – как второстепенная ($s_4^L = 0,5$).

Тогда оценка критерия качества L для альтернативы a_1 :

$$W_1^L = \sum_{i=1}^4 s_i^L w_{i1}^L = 1,625.$$

Проведя нормирование, получим:

$$\tilde{W}_1^L = W_1^L / n_L = 0,406.$$

Аналогичным образом определяются оценки других альтернатив:

$$\tilde{W}_2^L = 0,438, \tilde{W}_3^L = 0,750, \tilde{W}_4^L = 0,188, \tilde{W}_5^L = 0,292, \tilde{W}_6^L = 0,750, \tilde{W}_7^L = 0,375.$$

Этап 3. На данном этапе в соответствии с выражением (11) определяются комплексные оценки для альтернатив a_j :

$$W_1^1 = \tilde{W}_1^F + \tilde{W}_1^C + \tilde{W}_1^L = 1,531.$$

$$W_2^1 = 1,343, W_3^1 = 2,197, W_4^1 = 0,963, W_5^1 = 0,847, W_6^1 = 2,313, W_7^1 = 1,459.$$

Таким образом, расчет комплексных оценок для соответствующих альтернатив показал, что оптимальным вариантом из них является альтернатива a_3 . Тем не менее для формирования класса «Редакторы презентаций» только альтернативы a_3 будет недостаточно, так как в ней отсутствует функция, определенная как основная (функция f_6). Таким образом, необходимо осуществить выборку из имеющихся альтернатив по принципу максимальной (требуемой) функциональности минимальным количеством.

Этап 4. Для достижения данной цели методом Гомори решается задача целочисленного линейного программирования:

$$Z = 1,531x_1 + 1,343x_2 + 2,197x_3 + 0,963x_4 + 0,847x_5 + 2,313x_6 + 1,459x_7 \rightarrow \max,$$

$$\begin{cases} x_3 + x_4 + x_5 + x_7 \geq 1; \\ x_1 + x_2 + x_3 + x_6 + x_7 \geq 1; \\ x_1 + x_2 + x_3 + x_5 + x_6 + x_7 \geq 1; \\ x_1 + x_3 + x_4 + x_6 \geq 1; \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_7 \geq 1; \\ x_2 + x_4 + x_6 \geq 1; \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \leq 2; \\ 0 \leq x_i \leq 1, \quad i = \overline{1,7}. \end{cases}$$

Решая данную задачу, получим оптимальное решение: $X = (0, 0, 1, 0, 0, 1, 0)$, $Z(X) = 4,51$. Таким образом, класс «Редакторы презентации» для данного вуза будет определяться номенклатурой альтернатив a_3 и a_6 .

Для реализации и упрощения расчетов на всех этапах рассмотренного механизма, а также его внедрения в практическую деятельность образовательных организаций разработана автоматизированная информационная система «Комплект ПО» [14, 15].

Заключение

Предложенный механизм позволяет осуществить количественную оценку качества ПО в рамках его класса с последующим выбором наиболее рациональных для вуза программных средств. Такая оценка позволит упростить задачу по принятию обоснованного решения при формировании типового комплекта ПО, необходимого для реализации требований федеральных образовательных стандартов, и в целом повысит эффективность системы координации информатизации по данному направлению.

Литература

1. Бобровских, А.В. Перспективы использования прикладного программного обеспечения в образовательных организациях военной направленности / А.В. Бобровских // *Инновационные Технологии* –2019: сб. статей. – Пермь: Аэтерна, 2019. – С. 8–12.
2. Бобровских, А.В. Свободное программное обеспечение. Математические продукты / А.В. Бобровских, Т.Ю. Урывская, А.П. Алимов // *Инженерный вестник Дона (сетевое издание)*. – 2019. – № 9. – <http://www.ivdon.ru/ru/magazine/archive/N9y2019/6246>.
3. Бобровских, А.В. Перспективы использования программного обеспечения в образовательной деятельности курсантов ВУНЦ ВВС «ВВА» / А.В. Бобровских // *Молодежные чтения, посвященные памяти Ю.А. Гагарина: сб. статей*. – Воронеж: ВУНЦ ВВС «ВВА», 2019. – С. 23–27.
4. *Master sanctions chart, updated march 31, 2021*. – <https://www.state.gov/wp-content/uploads/2021/04/MASTER-Sanctions-chart-March-2021.pdf> (дата обращения: 06.09.2021).
5. Болотникова, О.В. Линейное программирование: симплекс-метод и двойственность / О.В. Болотникова, Д.В. Тарасов, Р.В. Тарасов. – Пенза: Изд-во ПГУ, 2015. – 84 с.
6. Шевченко, А.С. Методы оптимизации. Линейное программирование / сост. А.С. Шевченко. – Рубцовск: Рубцовский институт (филиал) АлтГУ, 2016. – 162 с.
7. Каштаева, С.В. Методы оптимизации / С.В. Каштаева. – Пермь: ИПЦ «Прокрость», 2020. – 84 с.
8. Гарина, С.В. Методы оптимизации и исследование операций / С.В. Гарина, М.Б. Никишин. – Саранск: Изд-во Мордовского университета, 2019. – 244 с.
9. Чернышова, Г.Д. Дискретные и вероятностные модели (Модели. Алгоритмы) / Г.Д. Чернышова, И.Н. Булгакова. – Воронеж: Издат. дом ВГУ, 2014. – 50 с.
10. ГОСТ 28195–89. Оценка качества программных средств. Общие положения. – Введ. 1990-07-01. – М.: ИПК Изд-во стандартов, 1989. – 31 с.
11. ГОСТ ИСО/МЭК 9126–2001. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. – Минск: Евразийский совет по стандартизации, метрологии и сертификации, 2001. – 13 с.
12. ГОСТ Р ИСО/МЭК 12119–2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование. – Введ. 2002-01-01. – М.: Госстандарт России: ИПК Изд-во стандартов, 2001. – 19 с.
13. Единый реестр российских программ для электронных вычислительных машин и баз данных. – <https://reestr.digital.gov.ru/> (дата обращения: 21.07.2021)
14. Бобровских, А.В. Автоматизированная информационная система для определения оптимального комплекта программного обеспечения в военных высших учебных заведениях / А.В. Бобровских, А.В. Гавриленко, А.А. Плахотин // *Молодежные чтения, посвященные памяти Ю.А. Гагарина: сб. статей*. – Воронеж: ВУНЦ ВВС «ВВА», 2021. – С. 8–14.
15. Свидетельство о государственной регистрации программы для ЭВМ № 2021664371. Российская Федерация. Автоматизированная информационная система для определения оптимального комплекта программного обеспечения «Комплект ПО» / А.В. Бобровских, А.В. Гавриленко, Ю.В. Бондаренко, А.П. Алимов; правообладатель А.В. Бобровских; опубл. 06.09.2021.

Бобровских Алексей Викторович, начальник научно-исследовательской лаборатории, Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия имени профессора Н.Е. Жуковского и Ю.А. Гагарина», г. Воронеж; fantom121128@mail.ru.

Бондаренко Юлия Валентиновна, д-р техн. наук, профессор, Воронежский государственный университет, г. Воронеж; bond.julia@mail.ru.

Поступила в редакцию 29 июля 2021 г.

FORMATION OF AN OPTIMAL SET OF SOFTWARE IN EDUCATIONAL ORGANIZATIONS

A.V. Bobrovskikh¹, fantom121128@mail.ru,

Yu.V. Bondarenko², bond.julia@mail.ru

¹ Military Scientific Educational Center of Military-Air Forces “N.E. Zhukovsky and Ju.A. Gagarin Military-Air Academy”, Voronezh, Russian Federation,

² Voronezh State University, Voronezh, Russian Federation

The problem of determining the nomenclature of the optimal set of application software (software) necessary for the implementation of educational programs in higher educational institutions (HEI) by quantifying its quality is considered. **Aim.** Development of a mechanism for a comprehensive assessment of the quality of software within the selected class, to simplify the task of making an informed decision when choosing a standard set of universities. **Materials and methods.** The proposed mechanism is based on the use of optimization methods and expert assessments. At the first stage of the mechanism, the basic requirements are clarified, taking into account the specifics of the work of the educational organization. A set of programs (alternatives) for the selected software class is formed, with the subsequent determination of the key functions characteristic of it on the basis of expert information. Further, according to the software quality indicators proposed in the work, they are evaluated for each alternative. Software quality indicators are a hierarchical three-level system (criterion, metric, evaluation element), in which the indicators of the higher levels are determined through the indicators of the lower levels. The integral assessment of the quality of an alternative is defined as the sum of the estimates of its quality criteria. At the final stage, the problem of integer linear programming is solved to determine the optimal range of alternatives within the corresponding software class, based on their conditions: maximizing the quality of an alternative containing the required set of functions, and minimizing their number. **Results.** The mechanism of determining the optimal alternatives for the formation of a standard set of software, in order to implement the necessary educational programs of the university, is considered. The main indicators of software quality and the mathematical apparatus for their assessment are formalized. **Conclusion.** The proposed mechanism allows for a quantitative assessment of the quality of software within its class, with the subsequent selection of the most rational software tools for the university. Such an assessment will simplify the task of making an informed decision when forming a standard set of software necessary for implementing the requirements of federal educational standards, and in general will increase the effectiveness of the system for coordinating informatization in this direction.

Keywords: software, alternative, quality indicators, quality assessment, standard kit, informatization of the university.

References

1. Bobrovskikh A.V. [Prospects for the use of applied software in military-oriented educational organizations]. *Innovatsionnyye Tekhnologii – 2019: sb. statey* [Innovative Technologies – 2019]. Perm, Aeterna Publ., 2019, pp. 8–12. (in Russ.)
2. Bobrovskikh A.V., Uryvskaya T.Yu., Alimov A.P. *Svobodnoye programmnoye obespecheniye. Matematicheskiye produkty* [Free software. Mathematical products]. *Inzhenernyy vestnik Dona*, 2019, no. 9. Available at: <http://www.ivdon.ru/ru/magazine/archive/N9y2019/6246>.
3. Bobrovskikh A.V. [Prospects for the use of software in the educational activities of cadets of the MERC AF “AFA”]. *Molodezhnyye chteniya, posvyashchennyye pamyati Yu.A. Gagarina: sb. statey* [Youth readings dedicated to the memory of Yuri Gagarin]. Voronezh, MERC AF “AFA”, 2019, pp. 23–27. (in Russ.)
4. Master sanctions chart, updated march 31, 2021. Available at: <https://www.state.gov/wp-content/uploads/2021/04/MASTER-Sanctions-chart-March-2021.pdf> (accessed 06.09.2021).
5. Bolotnikova O.V., Tarasov D.V., Tarasov R.V. *Lineynoye programmirovaniye: simpleks-metod i dvoystvennost'* [Linear programming: the simplex method and duality]. Penza, PSU Publ., 2015. 84 p.

6. Shevchenko A.S. *Metody optimizatsii. Lineynoye programmirovaniye* [Optimization methods. Linear programming]. Rubtsovsk, Rubtsovsky Institute (branch) of AltSU, 2016. 162 p.
7. Kashtaeva S.V. *Metody optimizatsii* [Optimization methods]. Perm, CPI “Prokrost”, 2020. 84 p.
8. Garina S.V., Nikishin M.B. *Metody optimizatsii i issledovaniye operatsiy* [Optimization methods and operations research]. Saransk, Mordovian Univ. Publ., 2019. 244 p.
9. Chernyshova G.D., Bulgakova I.N. *Diskretnyye i veroyatnostnyye modeli (Modeli. Algoritmy)* [Discrete and probabilistic models (Models. Algorithms)]. Voronezh, Publishing House of VSU, 2014. 50 p.
10. *GOST 28195–89*. [Evaluation of the quality of software tools. General provisions]. Moscow, Standartinform Publ., 1989. 31 p. (in Russ.)
11. *GOST ISO/IEC 9126–2001*. [Information technology. Evaluation of software products. Quality characteristics and guidelines for their use]. Minsk, Eurasian Council for Standardization, Metrology and Certification, 2001. 13 p. (in Russ.)
12. *GOST R ISO/IEC 12119–2000*. [Information technology. Software packages. Quality requirements and testing]. Moscow, Gosstandart of Russia, IPK Standartinform Publ., 2001. 19 p. (in Russ.)
13. *Edinyy reyestr rossiyskikh programm dlya elektronnykh vychislitel'nykh mashin i baz dannykh* [Unified Register of Russian programs for electronic computers and databases]. Available at: <https://reestr.digital.gov.ru/> (accessed 21.07.2021).
14. Bobrovskikh A.V., Gavrilenko A.V., Plakhotin A.A. [Automated information system for determining the optimal set of software in military higher educational institutions]. *Molodezhnyye chteniya, posvyashchennyye pamyati Yu.A. Gagarina: sb. statey* [Youth readings dedicated to the memory of Yuri Gagarin]. Voronezh, MERC AF “AFA”, 2021, pp. 8–14. (in Russ.)
15. Bobrovskikh A.V., Gavrilenko A.V., Bondarenko Yu.V., Alimov A.P. *Avtomatizirovannaya informatsionnaya sistema dlya opredeleniya optimal'nogo komplekta programmnoy obespecheniya “Komplekt PO”* [Automated information system for determining the optimal software package “Software Kit”]. Certificate of state registration of the computer program no. 2021664371; publ. 06.09.2021.

Received 29 July 2021

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Бобровских, А.В. Формирование оптимального комплекта программного обеспечения в образовательных организациях / А.В. Бобровских, Ю.В. Бондаренко // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2021. – Т. 21, № 4. – С. 5–18. DOI: 10.14529/ctcr210401

FOR CITATION

Bobrovskikh A.V., Bondarenko Yu.V. Formation of an Optimal Set of Software in Educational Organizations. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2021, vol. 21, no. 4, pp. 5–18. (in Russ.) DOI: 10.14529/ctcr210401