

ОЦІНКА "ПРАКТИЧНОСТІ" ТА "КОРЕКТНОСТІ" СПЕЦІАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ВОЄННОГО ПРИЗНАЧЕННЯ

В статті проведено оцінку "практичності" та "коректності" спеціального програмного забезпечення автоматизованих інформаційних систем воєнного призначення. Показано, що інформаційний ресурс високоточної зброї повинний мати повний набір програмних засобів як активного і пасивного захисту від атак на його інформаційні системи, так і його активних і пасивних впливів стосовно всіх існуючих і перспективних систем озброєння противника. Серед проблем, що пов'язані із створенням спеціального програмного забезпечення (СПЗ) автоматизованих інформаційних систем воєнного призначення, можливо виділити дві основні: проблема створення якісного спеціального програмного забезпечення; проблема раціоналізації ефективності праці учасників життєвого циклу СПЗ. Одним з перспективних напрямків їх вирішення є реалізація регламентованого технологічного процесу. Системний полягає в скороченні помилок у програмному забезпеченні шляхом їх своєчасного виявлення та локалізації, зменшення трудомісткості їх виявлення та виправлення за рахунок чітких і упорядкованих структури та зв'язків.

Характеристикою практичності СПЗ є вивчаємість, що характеризується (зусиллями, необхідними для освоєння користувачами умов, процедур і правил застосування ПЗ). Вони описуються наступними показниками: середній час освоєння програмного виробу обслуговуючим персоналом, коефіцієнт повноти демонстраційної версії, коефіцієнт повноти та гнучкості довідкової системи.

Характеристика зручність експлуатації та коректність СПЗ характеризується легкістю підготовки вхідних даних і запуску в роботу СПЗ. Кількісна оцінка характеризується такими показниками: коефіцієнт ступеню автоматизації контролю вводу даних, коефіцієнт використання ефективних засобів введення даних. Оцінювання необхідних інтелектуальних зусиль для створення програми характеризується кількістю необхідних елементарних рішень при створення програмного коду, однак він не враховує дії по відлагодженню програми, тому автори пропонують ввести коефіцієнт реальної складності, що полягає в оцінці витрат на прийняття готової програми. Наведені метрики дозволяють отримати їх чисельні значення, на основі яких можливо порівнювати програми за цими характеристиками якості.

Ключові слова: спеціальне програмне забезпечення, автоматизовані інформаційні системи, оцінка практичності та коректності, адекватність та достовірність рівня якості програмування.

Вступ та постановка задачі. Сьогодні спостерігається значний зріст науково-технічних досліджень в напрямку інформатизації і автоматизації управління військами і зброєю. На прикладі розвитку Збройних сил провідних країн світу треба очікувати автоматизації всіх рівнів їх організаційних структур. Нажаль в Україні у перехідний період інформаційно-аналітичне забезпечення поки що зберігається як один з видів забезпечення всіх інших видів боротьби. Вже після завершення перехідного періоду, інформаційно-аналітичне забезпечення поступове вийде за межі виду, що забезпечує, і стане бойовим, тобто придбає самостійний характер серед багатьох інших форм і способів забезпечення військ (сил) головним чином буде досягатися через перевагу в одержанні достовірної інформації, мобільності, швидкості реакції, у точному вогневому й інформаційному впливі в реальному масштабі часу по численних об'єктах його економіки, військових об'єктах стратегічних та тактичних діяч і при мінімально можливому ризику для своїх сил і засобів.

При цьому буде потрібна максимальна інтеграція механізмів обробки і аналізу інформації багатofункціональних ударних і оборонних систем на основі автоматичних пристроїв управління та за рахунок значного зниження кількості рівнів управління військами, силами, засобами. Також буде потрібна надійна захищеність як окремих ударних і оборонних елементів високоточних систем, так і стратегічної системи в цілому від впливу всіх видів сучасного інформаційного впливу.

Інформаційний ресурс високоточної зброї повинний мати повний набір програмних засобів як активного і пасивного захисту від атак на його інформаційні системи, так і його активних і пасивних впливів стосовно всіх існуючих і перспективних систем озброєння противника.

Серед проблем, що пов'язані із створенням спеціального програмного забезпечення (СПЗ) автоматизованих інформаційних систем воєнного призначення, можливо виділити дві основні:

- проблема створення якісного спеціального програмного забезпечення;
- проблема раціоналізації ефективності праці учасників життєвого циклу СПЗ;

Одним з перспективних напрямків їх вирішення є реалізація детально регламентованого технологічного процесу. При цьому необхідний рівень регламентації може бути досягнутий в результаті системного підходу до забезпечення заданих характеристик якості СПЗ на різних стадіях життєвого циклу програмного забезпечення. Суть системного підходу відповідно до забезпечення цільової якості СПЗ полягає в скороченні помилок у програмному забезпеченні шляхом їх своєчасного виявлення та локалізації, зменшення трудомісткості їх виявлення та виправлення за рахунок чітких і упорядкованих структури та зв'язків.

Значної частини труднощів, що виникають при розробці та впровадженні СПЗ воєнного призначення, можливо уникнути, якщо з самого початку створити систему спеціального програмного забезпечення у відповідності з певною методологією. Ця методологія повинна враховувати безперервність процесів розробки та впровадження, статичність цілей розробника, динамічність вимог замовника, необхідність забезпечення високої надійності та точності функціонування, зручності експлуатації та гнучкості системи при зміні нормативних даних та інформаційних масивів.

Крім критичності СПЗ ВП в плані правильного, надійного та стійкого функціонування, витрати на доробку вже діючих програм та усунення помилок, що виявляються в процесі використання, досягають іноді 75% загальної суми експлуатаційних витрат, [1,2]. За іншими даними [1, 3] на це витрачається від 30 до 50% загального бюджету розробки, а помилки, що не виявлені на початку життєвого циклу, коштують від 70 до 85% вартості переробки.

Як показано на рис. 1, набагато дорожче виправити помилки, які знайдено пізніше в проекті [3].

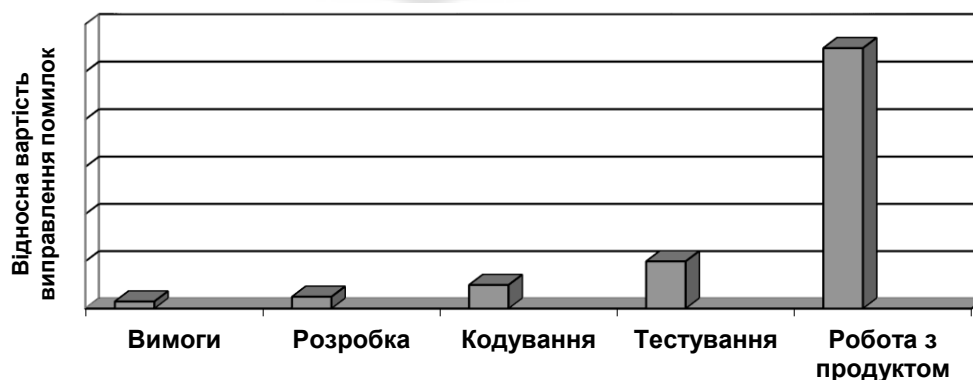


Рисунок 1 – Відносна вартість виправлення помилок в залежності від етапу, на якому вони виявлені

У 1994 році Консалтингова група фірми IBM провела дослідження стану справ у 24 провідних компаніях США, що розробляють великі програмні системи. В її звіті [4] як і у статті [5] зазначалося, що:

- у 55% проектів перевищено заплановану вартість розробки;
- у 68% проектів порушено договірні строки розробки;
- 88% проектів довелося суттєво переробляти.

Аналогічні дослідження були проведені Standish Group у 1994 році по 8380 програмним проектам, що розроблялись у державному та приватному секторах США. Ці дослідження [6], а також аналіз вітчизняних досліджень [7] показують, що:

- 31% всіх програмних проектів закриваються до їх повного закінчення;
- 53% успішно закінчених проектів в середньому на 18% перевищують заплановану вартість;
- з цих 53% проектів тільки 42% забезпечують в програмних продуктах всі властивості та функції, що планувалися;
- тільки 9% проектів закінчується у встановлений термін і не перевищують заплановані для проекту кошти.

В своїх дослідженнях Standish Group виділяє наступні критерії успішності виконання проектів у розглянутій області:

Таблиця 1

№	Критерій успіху	Ступінь важливості
1.	Участь користувачів у проекті	19%
2.	Належне управління на виконавчому рівні	16%
3.	Ясні вимоги до розроблювального продукту	15%
4.	Кваліфіковане планування	11%
5.	Реалістичні очікування результатів	10%
6.	Більш частий контроль виконання етапів плану	9%
7.	Компетентність персоналу	8%
8.	Чіткий розподіл обов'язків і прав у команді виконавців	6%
9.	Чітка постановка цілей і їхня відповідність стратегічним задачам організації	3%
10.	Мотивація персоналу на напружену роботу	3%

Можна зробити висновок, що більшість перерахованих критеріїв відноситься до області управління проектами [8].

Тому в циклі робіт авторами вирішується важлива для теорії і практики діяльності Збройних Сил та держави в цілому наукова проблема розробки науково-методичних основ гарантування та оцінки цільової якості спеціального програмного забезпечення автоматизованих інформаційних систем воєнного призначення.

Складовими цієї проблеми є:

1. Створення системи гарантування цільової якості СПЗ.
2. Формалізація процесу оцінки якості СПЗ.
3. Розробка пропозицій щодо організаційно-штатної структури та алгоритму роботи підрозділу, який займається розробкою спеціального програмного забезпечення.

Мета даної статті. Оцінка "практичності" та "коректності" спеціального програмного забезпечення автоматизованих інформаційних систем воєнного призначення.

Основні результати дослідження. Зупинимось на основних поняттях кваліметрії та інженерії програмного забезпечення, якими будемо оперувати у подальшому. *Якість СПЗ* – це сукупність властивостей (атрибутив), що визначають його корисність для користувачів у відповідності з функціональним призначенням та висунутими вимогами. *Характеристика якості* – поняття, що відображає окремі чинники, які впливають на якість програм та є такими,

що можуть бути виміряні. *Критерій якості* – чисельний показник, що характеризує ступінь, в якій програмі властиві атрибути якості, що оцінюються. Критерій якості повинен відповідати таким вимогам: чисельно характеризувати основну цільову функцію програми; забезпечувати можливість визначення витрат, необхідних для досягнення потрібного рівня якості, а також ступеню впливу на показник якості різноманітних зовнішніх чинників; бути по можливості простим, добре вимірюватись та мати малу дисперсію. Для вимірювання характеристик і критеріїв якості СПЗ використовуються метрики. *Метрика якості* – це система оцінок якості СПЗ. Ці оцінки можуть здійснюватись на рівні критеріїв якості СПЗ, або на рівні окремих характеристик. При першому підході система оцінок дозволяє порівнювати програмне забезпечення за якістю безпосередньо. При цьому оцінки не можуть бути проведені без суб'єктивних вимірів властивостей СПЗ. При другому підході оцінку характеристик СПЗ можливо здійснити об'єктивно і достовірно, але оцінювання якості СПЗ в цілому буде пов'язана із суб'єктивною інтерпретацією отриманих оцінок.

Стандартами, які прийняті в області якості програмного забезпечення в Україні, характеристика якості програмного забезпечення *usability* перекладається як зручність чи простота використання. В той же час однією з її підхарактеристик є *зручність експлуатації та обслуговування*, що не є коректним з точки зору семантики поняття цієї характеристики. Пропонується використовувати поняття "практичність" як інтегральну характеристику більш високого рівня ієрархії ніж "зручність". Під практичністю СПЗ будемо розуміти його здатність бути зрозумілим, вивчаємим, зручним в експлуатації та обслуговуванні при використанні в заданих умовах. Однією з підхарактеристик практичності є *зрозумілість* СПЗ – його здатність надавати можливість користувачу зрозуміти, чи підходить йому даний програмний продукт, і як його застосовувати для конкретних завдань і умов використання.

Як найбільш просту метрику зрозумілості СПЗ пропонується використовувати оцінку рівня коментованості програми, L_{com} :

$$L_{com} = \frac{N_{com}}{N_{line}}, \quad (1)$$

де – N_{com} - кількість коментарів в програмі;

N_{line} - загальна кількість рядків або операторів вихідного тексту програми.

Отже, метрика L_{com} відображає насиченість програми коментарями. Виходячи з практичного досвіду будемо вважати, що $L_{com} \geq 0.1$, тобто на кожні десять рядків або операторів програми має бути мінімум один коментар. Але, як свідчить аналіз, дуже часто коментарі розподіляються у тексті програми нерівномірно: на початку програми спостерігається їх надлишок, а в середині чи в кінці – їх бракує. Це пояснюється, очевидно, тим, що на початку програми, як правило, розміщені оператори опису ідентифікаторів, які потребують більш ретельного коментування. Крім того, на початку програми також розташовані блоки з інформацією про виконавця, функціональне призначення програми та ін. Така насиченість компенсує брак коментарів у тілі програми і тому формула (1) недостатньо адекватно відображає коментованість функціональної частини тексту програми. Пропонується варіант, коли вся програма поділяється на n рівних сегментів і для кожного з них окремо визначається $L_{com}^{(i)}$:

$$L_{com}^{(i)} = \text{sign}\left(\frac{N_{com}^{(i)}}{N_{line}^{(i)}} - 0.09\right). \quad (2)$$

При цьому рівень коментованості всієї програми, L_{com} буде визначатись, як:

$$L_{com} = \sum_{i=1}^n L_{com}^{(i)}. \quad (3)$$

Виходячи з мінімально достатньої кількості коментарів відносно кількості рядків або операторів програми ($L_{com} \geq 0.1$) мінімально достатній рівень коментованості програми будемо вважати досягнутим, якщо виконується умова $L_{com} = n$. В іншому випадку i -й фрагмент програми, для якого $L_{com}^{(i)} < 0$, доповнюється коментарями до номінального рівня.

Наступною підхарактеристикою практичності СПЗ є вивчаємість. Стандартом [9] ця підхарактеристика практичності СПЗ характеризується "зусиллями, необхідними для освоєння користувачами умов, процедур і правил застосування ПЗ". Для її кількісної оцінки пропонуються наступні показники:

1. *Середній час освоєння програмного виробу* обслуговуючим персоналом, $\overline{T_{fam}}$ (статистичний показник). Мінімальне та максимальне значення цього показника повинно визначатись на етапі формулювання специфікацій на програмний продукт в процесі спілкування представників замовника та виконавця (можливо в формі тестування):

$$\overline{T_{fam}} = \frac{\sum_{i=1}^n T_i}{n}, \quad (4)$$

де T_i - час освоєння програмного засобу i -м користувачем;

n - кількість користувачів, які вивчали правила користування програмним засобом.

2. *Коефіцієнт повноти демонстраційної версії* K_{DV} , який відображає повноту та детальність ілюстрації функцій СПЗ та об'єктів, над якими вони виконуються:

$$K_{DV} = \frac{\sum_{i=1}^n N_i^{(func)}}{n}, \quad (5)$$

де $N_i^{(func)}$ - кількість функцій СПЗ, демонстрація виконання яких доступно і повно представлена у демоверсії;

n - загальна кількість функціональних блоків в СПЗ.

3. *Коефіцієнт повноти та гнучкості довідкової системи* K_{help} , одиничними показниками ступеню реалізації якого є такі:

наявність загальних відомостей про систему (0, 1);

можливість організації тематичного пошуку в загальній довідковій системі (0, 1);

можливість доступу при будь-якому режимі роботи до повної довідки за допомогою командної кнопки чи комбінації керуючих клавіш (0, 1);

використання контекстної довідки (0, 1);

варіанти роботи з даними довідки (друкування чи занесення в файл), (0, 1);

$$K_{help} = \frac{\sum_{i=1}^5 N_i^{(index)}}{5}, \quad (6)$$

де $N_i^{(index)}$ - одиничні показники ступеню реалізації повноти та гнучкості довідкової системи.

Підхарактеристика зручність експлуатації та обслуговування характеризується "легкістю підготовки вхідних даних і запуску в роботу ПЗ, а також необхідними умовами інтерфейсу користувача в процесі функціонування ПЗ" [9]. Для її кількісної оцінки пропонуються наступні показники:

1. Коефіцієнт ступеню автоматизації контролю вводу даних, K_{avtinp} :

$$K_{avtinp} = \frac{V_{avt}}{V}, \quad (7)$$

де – V_{avt} - обсяг початкових даних, під час введення яких здійснюється контроль їх правильності реалізованими у програмному виробі засобами;

V - загальний обсяг вхідних даних, які необхідно перевіряти.

2. Коефіцієнт використання ефективних засобів введення даних, K_{effinp} :

$$K_{effinp} = \frac{V_{eff}}{V_{inp}}, \quad (8)$$

де – V_{eff} - обсяг початкових даних, для вводу яких застосовуються автоматизовані засоби;

V_{inp} - загальний обсяг початкових даних, що вводиться.

Коректність СПЗ міжнародним стандартом ISO 9126-1 [10] трактується як спроможність програмного засобу забезпечувати правильні та прийнятні для користувача результати та зовнішні ефекти.

Наступні п'ять характеристик базуються на метриці М.Холстеда V^* , за допомогою якої описується потенційний об'єм коду, який відповідає максимально компактному тексту програми, що реалізує даний алгоритм [11]:

$$V^* = n^* \log_2(n^*). \quad (9)$$

Для оцінки теоретичної довжини програми, L_t М.Холстед вводить апроксимуючу формулу:

$$L_t = n_1 \log_2(n_1) + n_2 \log_2(n_2), \quad (10)$$

де n_1 – кількість унікальних операторів програми (словник операторів);

n_2 – кількість унікальних операндів програми (словник операндів).

Фізичний зміст цієї оцінки базується на основних концепціях теорії інформації, за аналогією з якими частота використання операторів та операндів в програмі пропорційна двійковому логарифму кількості їх типів. Вираз (10) являє собою ідеалізовану апроксимацію виразу (9), тобто справедливий для потенційно коректних програм, які не мають надмірності та недосконалостей (стилістичних помилок). М.Холстед [11] стверджує, що для стилістично коректних програм відхилення в оцінці теоретичної довжини L_t від реальної L не перевищує 10%. Пропонується застосовувати L_t як еталонне значення довжини програми із словником n . Вимірюючи n_1 (кількість унікальних операторів програми), n_2 (кількість унікальних операндів програми), N_1 (загальну кількість операторів у програмі) та N_2 (загальну кількість операндів у програмі) та порівнюючи L_t з L для певної програми, при більш, ніж 10%-вому відхиленні можливо зробити висновок про наявність в програмі стилістичних помилок, тобто недосконалостей.

В якості наступної оцінки, яка належить до метрик коректності СПЗ, згідно М.Холстеду, пропонується застосовувати коефіцієнт рівня якості програмування, K_{LQ} [12]:

$$K_{LQ} = \frac{V^*}{V}, \quad (11)$$

де – V^* - розраховується за формулою (9);

V - фактичний обсяг програми $V = N \log_2(n_1 + n_2)$ (біт).

Під бітом розуміється логічна одиниця інформації – символ, оператор, операнд.

Вихідною посилкою для введення цієї оцінки є припущення про те, що при зниженні стилістичної якості програмування зменшується змістовне навантаження на кожний компонент програми і, як наслідок, збільшується фактичний обсяг реалізації вихідного алгоритма. Враховуючи це, можливо оцінити якість програмування на основі ступеня

збільшення фактичного обсягу програми відносно потенційного V^* . Очевидно, що для ідеальної програми $K_{LQ} = 1$, а для реальної – завжди $K_{LQ} < 1$.

Для підвищення адекватності та достовірності коефіцієнту рівня якості програмування М.Холстед пропонує апроксимувати цю оцінку виразом, що включає тільки фактичні параметри, тобто параметри реальної програми: $K_{LQ}^{\wedge} = \frac{2n_2}{n_1 N_2}$. Доцільність цього пояснюється

тим, що список параметрів програми може бути штучно розширений (оскільки залежить від рівня реалізації вихідного алгоритму), що, в свою чергу, веде до збільшення метрики рівня якості програмування. На основі K_{LQ}^{\wedge} М.Холстед вводить характеристику I , яку трактує як інтелектуальний зміст конкретного алгоритма, інваріантний відносно мови реалізації:

$$I = K_{LQ}^{\wedge} V. \quad (12)$$

Перетворюючи (12) з урахуванням (11), отримаємо: $I = K_{LQ}^{\wedge} V = K_{LQ} V = \frac{V^* V}{V} = V^*$.

Еквівалентність I та V^* очевидно свідчить про те, що ми маємо справу з характеристикою інформативності програми. Введення характеристики I дозволяє оцінити розумові витрати на створення програми. Процес створення програми умовно можливо розділити на два етапи: осмислення алгоритму; перетворення алгоритму в терміни мови програмування, тобто пошук в словнику мови відповідної конструкції, її змістовне наповнення і запис. Використовуючи цю формалізацію в методиці М.Холстеда, можливо стверджувати, що написання програми у відповідності з відомим алгоритмом є L_t -кратна вибірка операторів і операндів із словника програми n , причому кількість порівнянь складе $\log_2(n)$. Якщо врахувати, що кожна вибірка порівняння містить, в свою чергу, певну кількість розумових елементарних рішень, то можливо поставити у відповідність змістовному навантаженню кожної конструкції програми кількість та складність цих елементарних розумових рішень. Кількісно це можливо характеризувати за допомогою K_{LQ} , оскільки $\frac{1}{K_{LQ}}$ можливо використовувати як середній коефіцієнт складності, що впливає на швидкість вибірки для даної програми. За таких умов оцінювання необхідних інтелектуальних зусиль щодо створення програми, K_{int} може бути визначена наступним чином:

$$K_{\text{int}} = L_t \log_2 \left(\frac{n}{K_{LQ}} \right). \quad (13)$$

Таким чином, K_{int} характеризує кількість необхідних елементарних рішень при створенні програмного коду. Але K_{int} адекватно характеризує тільки початкові зусилля по створенню програми, оскільки цей коефіцієнт не враховує дії щодо відлагодження програми, які потребують інтелектуальних зусиль іншого характеру. Тому пропонується ввести коефіцієнт реальної складності, $K_{\text{int}}^{\text{real}}$, фізичний зміст якого полягає в оцінці не інтелектуальних витрат на розробку програми, а витрат на сприйняття готової програми. При цьому замість теоретичної довжини програми L_t використовується її реальна довжина L :

$$K_{\text{int}}^{\text{real}} = L \log_2 \left(\frac{n}{K_{LQ}} \right). \quad (14)$$

Перетворюючи формулу (13) з урахуванням виразів для V^* та V , отримаємо $K_{\text{int}} = \frac{V^2}{V^*}$.

Таке подання K_{int} наглядно ілюструє доцільність розбиття програми на окремі модулі,

оскільки інтелектуальні витрати виявляються пропорційними квадрату обсягу програми, який завжди більше суми квадратів обсягів окремих модулів.

Висновки. Таким чином, наведені метрики, які пропонується застосовувати для оцінки практичності та коректності СПЗ АІС ВП, дозволяють отримати їх чисельні значення, на основі яких можливо порівнювати програми за цими характеристиками якості. Крім того, розглянуті метрики базуються на аналізі вихідних текстів програм, що забезпечує єдиний підхід до автоматизації їх обчислення.

ЛІТЕРАТУРА

1. Boehm V.W., Philip N. Papaccio. Understanding and Controlling Software Costs. IEEE Transactions on Software Engineering, №14(10), 1988, P. 1462-1476.
2. Черников Б.В., Поклонов Б.Е. Управление качеством программного обеспечения. Практикум. М.: НД «Форум», 2012, 240 с.
3. Grady R.B. An economic release decision model: insights into software project management // In proceedings of the applications of software measurement conference, Orange Park, FL: Software Quality Engineering, 1999. - P. 227-239.
4. Barlas S. Anatomy of a Runaway: What Grounded the AAS // IEEE Software, January 1996. - P. 104-106.
5. Наумов А.И., Взоров В.Н. Концепция управления знаниями и практика компании. Вестник Московского университета. Серия 24. Менеджмент. – 2012. - №2 – С. 33 – 78.
6. Gibbs W. Software's Chronic Crisis // Scientific America, September 1994. - P. 86-95.
7. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Г.М. Коротун, В.Ю. Суслов: НАН Украины, Институт программных систем. - К.: Академперіодика, 2002. - 503с.
8. Липаев В.В. Выбор и оценивание характеристик качества программных средств. Методы и стандарты. М.: СИНТЕГ, 2001, 228 с.
9. ДСТУ 2850-94 Програмні засоби ЕОМ. Показники і методи оцінювання якості
10. ISO/IEC 9126-1. 2001. Software engineering – Software product quality – Part 1: Quality model. Geneva, Switzerland: International Organization for Standardization;
11. Молодцова О.П. Управління якістю програмної продукції: Навч. посібник // Київський національний економічний університет. – К.: КНЕУ, 2001. – 248с.
12. Холстед М.Х. Начало науки о программах. – М.: Финансы и статистика, 1981. – 128с.

REFERENCES:

1. Boehm V.W., Philip N. Papaccio. Understanding and Controlling Software Costs. IEEE Transactions on Software Engineering, №14(10), 1988, pp. 1462-1476.
2. Chernikov B.V., Poklonov B.E. (2012), Upravlenie kachestvom programmnoho obespechenija. Praktikum. Moskva.: ND «Forum», 240 p.
3. Grady R.B. An economic release decision model: insights into software project management // In proceedings of the applications of software measurement conference, Orange Park, FL: Software Quality Engineering, 1999. P. 227-239.
4. Barlas S. Anatomy of a Runaway: What Grounded the AAS // IEEE Software, January 1996. - P. 104-106.
5. Naumov A.I., Vzorov V.N. (2012), Konceptija upravljenija znanijami i praktika kompanii. Vestnik Moskovskogo universiteta. Serija 24. Menedzhment. №2, S. 33 – 78.
6. Gibbs W. Software's Chronic Crisis // Scientific America, September 1994. - P. 86-95;
7. F.I. Andon, G.I. Koval', G.M. Korotun, V.Ju. Suslov (2002), Osnovy inzhenerii kachestva programmnyh sistem / NAN Ukrainy, Institut programmnyh sistem. K.: Akademperiodika, 503s.
8. Lipaev V.V. (2001), Vybor i ocenivanie harakteristik kachestva programmyh sredstv. Metody i standarty. M.: SINTEG, 228 s.
9. DSTU 2850-94 Programni zasoby EOM. Pokaznyky i metody ocinjuvannja jakosti;
10. ISO/IEC 9126-1. 2001. Software engineering – Software product quality – Part 1: Quality model. Geneva, Switzerland: International Organization for Standardization;
11. Molodcova O.P. (2001), Upravlinnja jakistju programnoi' produkcii': Navch. posibnyk // Kyi'vs'kyj nacional'nyj ekonomichnyj universytet. – K.: KNEU, 248s.
12. Holsted M.H.(1981), Nachalo nauki o programmah. M.: Finansy i statistika, 128s.

D.Sci. Tech. Lienkov S.V., PhD Gryschak O., PhD Zhyrov G., Phd Pampukha I.
**ASSESSMENT OF "PRACTICALITY" AND "CORRECTNESS" OF SPECIAL SOFTWARE OF
AUTOMATED MILITARY INFORMATION SYSTEMS**

The assessment of the "practicality" and "correctness" of special software of the automated military information systems is considered in the article. It's shown that the information resource of the high-precision weapons must have a full set of software as active and passive protection against attacks on its information systems, and its active and passive influences on all existing and promising weapons systems. Among the problems associated with the creation of special software (SS) of the automated information military systems, there are two main ones: the problem of creating high-quality special software; the problem of rationalizing the efficiency of the participants of the SS life cycle. One of the promising areas of their solution is the implementation of the regulated technological process. The systematic approach is to reduce software errors by detecting and locating them in the timely manner, reducing the complexity of detecting and correcting them through clear and orderly structures and connections.

A characteristic of the SS's practicality is the studied, that is characterized by efforts required for users to master the conditions, procedures and rules of application of the software. They are described by the following indicators: the average time of development of the software product by service personnel, the completeness coefficient of the demo version, the completeness and flexibility coefficient of the help system.

The characteristics of ease operation and correctness of SS is characterized by ease of preparation of input data and start-up of SS. The quantitative assessment is characterized by the following indicators: the coefficient of the degree of automation of data entry control, the coefficient of use of effective means of data entry. The estimating of the necessary intellectual effort to create a program is characterized by the number of necessary basic solutions when creating program code, but it doesn't take into account debugging actions, so the authors propose to introduce a factor of real complexity, that is to estimate the cost of program perception. The given metrics allow to receive their numerical values on the basis of that it's possible to compare programs on these quality characteristics.

Keywords: special software, automated information systems, assessment of practicality and correctness, adequacy and reliability of the programming quality level.

