

О. С. ИСАКОВ, О. Ю. ЧЕРЕДНИЧЕНКО, В. В. МОЗГІН, О. В. ЯНГОЛЕНКО

ДОСЛІДЖЕННЯ МОДЕЛЕЙ ПРОЦЕСІВ ТЕСТУВАННЯ ЗРУЧНОСТІ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ

У статті розглянуто питання тестування зручності використання програмного забезпечення (ПЗ). Для навантажувального тестування був розроблений тест-план. Результати навантажувального тестування показали, що розроблене ПЗ працює стабільно при ста одночасно працюючих користувачах, про що свідчить нульовий відсоток помилок та невеликий час відповіді системи на запити користувачів. Для дослідження зручності використання розробленого ПЗ у межах стратегії реального світу використано метод опитування. Це рішення обумовлено низькою вартістю дослідження та ненав'язливістю даного методу. Опитування надає можливість обробки великого обсягу статистичних даних, що підвищує надійність вимірювання. Для проведення дослідження зручності використання розробленого ПЗ використано метод опитування користувачів Software Usability Measurement Inventory (SUMI). Опитування вимірює такі аспекти задоволення користувачів: ефективність; емоційна реакція; здатність допомагати; контроль; засвоєваність. Для отримання надійних результатів було опитано 12 користувачів за допомогою анкети SUMI. Їхні відповіді були інтерпретовані та оброблені на основі моделі Раша. В результаті були отримані оцінки зручності використання програмних рішень.

Ключові слова: тестування, програмне забезпечення, навантажувальне тестування, опитування, зручність використання ПЗ.

A. S. ISAKOV, O. YU. CHEREDNICHENKO, V. V. MOZGIN, O. V. YANGOLENKO

ИССЛЕДОВАНИЕ МОДЕЛЕЙ ПРОЦЕССОВ ТЕСТИРОВАНИЯ УДОБСТВА ИСПОЛЬЗОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ

В статье рассмотрены вопросы тестирования удобства использования программного обеспечения (ПО). Для нагрузочного тестирования был разработан тест-план. Результаты нагрузочного тестирования показали, что разработанное ПО работает стабильно при ста одновременно работающих пользователях, о чем свидетельствует нулевой процент ошибок и небольшое время ответа системы на запросы пользователей. Для исследования удобства использования разработанного ПО в рамках стратегии реального мира использован метод опроса. Это решение обусловлено низкой стоимостью исследования и ненавязчивостью данного метода. Опрос предоставляет возможность обработки большого объема статистических данных, что повышает надежность измерения. Для проведения исследования удобства использования разработанного ПО использован метод опроса пользователей Software Usability Measurement Inventory (SUMI). Опрос измеряет такие аспекты удовлетворения пользователей как: эффективность; эмоциональная реакция; способность помогать; контроль; усвояемость. Для получения надежных результатов было опрошено 12 пользователей с помощью анкеты SUMI. Их ответы были интерпретированы и обработаны на основе модели Раша. В результате были получены оценки удобства использования программных решений.

Ключевые слова: тестирование, программное обеспечение, нагрузочное тестирование, опрос, удобство ПО.

A. S. ISAKOV, O. YU. CHEREDNICHENKO, V. V. MOZGIN, O. V. YANGOLENKO

TESTING MODELS RESEARCH OF THE USABILITY OF SOFTWARE PRODUCTS

The article deals with the issues of testing the software usability. Usability evaluation is an urgent problem nowadays as it is a determinative factor of software success among its users. This problem should be solved during the whole software development lifecycle in order to create the best product for consumers. Also, usability evaluation should be conducted after software delivering to the end users in order to estimate the degree of their satisfaction and to notice what has to be changed in the next product's version. A test plan was developed for load testing in the paper. The results of load testing showed that the developed software works stably with a hundred of simultaneously working users; it was confirmed by a zero percentage error and a small response time of the system to user's inquiries. The survey method was used to study the convenience of using the developed software within the real-world strategy. This decision was taken due to the low cost of research and the unobtrusive nature of this method. The survey provides the ability to process a large amount of statistical data, which increases the reliability of the measurement. The Software Usability Measurement Inventory (SUMI) survey method was used to measure the usability of the developed software. In order to study the usability of the developed software, it was performed the usability comparing of the developed system with similar systems. The plurality of software products was compared. Then several users evaluated the usability of the proposed software implementing the SUMI method. Based on individual assessments a group of experts estimated each product and compared its usability. The survey measured the following aspects of user satisfaction: efficiency; emotional reaction; ability to help; control; accessibility. In order to receive the reliable results, 12 users were polled using the SUMI questionnaire. Their responses were interpreted and processed on the basis of the Rash model. As a result, usability estimates were obtained. The consistency of expert judgments is determined with the help of the coefficient of variation.

Keywords: testing, software, load testing, polling, software usability.

Вступ. Тестування програмного забезпечення є важливим етапом життєвого циклу виробництва програмних систем. Міжнародний стандарт ISO 9126 [1] визначає оціночні характеристики якості програмного забезпечення. Поняття якості програмного забезпечення включає різні характеристики. Кожна модель відображення якості програмного забезпечення включає різне число рівнів ієрархії та загальне число характеристик якості. Різні автори створили різні моделі якості зі своїм набором

характеристик та атрибутів. Ці моделі є корисними для планування та оцінки якості програмних продуктів.

ISO/IEC визначає три взаємопов'язані моделі якості програмного забезпечення (ISO 9126-01 Software Engineering – Product Quality, Part 1: Quality Model [1]) – внутрішня якість, зовнішня якість та якість в процесі експлуатації, а також набір відповідних робіт з оцінки якості програмного забезпечення (ISO14598-98 Software Product

Evaluation).

Стандарт поділяється на чотири частини, що описують наступні питання: модель якості; зовнішні метрики якості; внутрішні метрики якості; метрики якості у використанні. Згідно цього стандарту модель якості класифікує якість програмного забезпечення у

шести структурних наборах характеристик, що мають свої атрибути (рис. 1). Короткий опис семантики характеристик моделі якості наведено у табл. 1. Опис атрибутів характеристик якості програмного забезпечення наведено у таблицях 2–7.

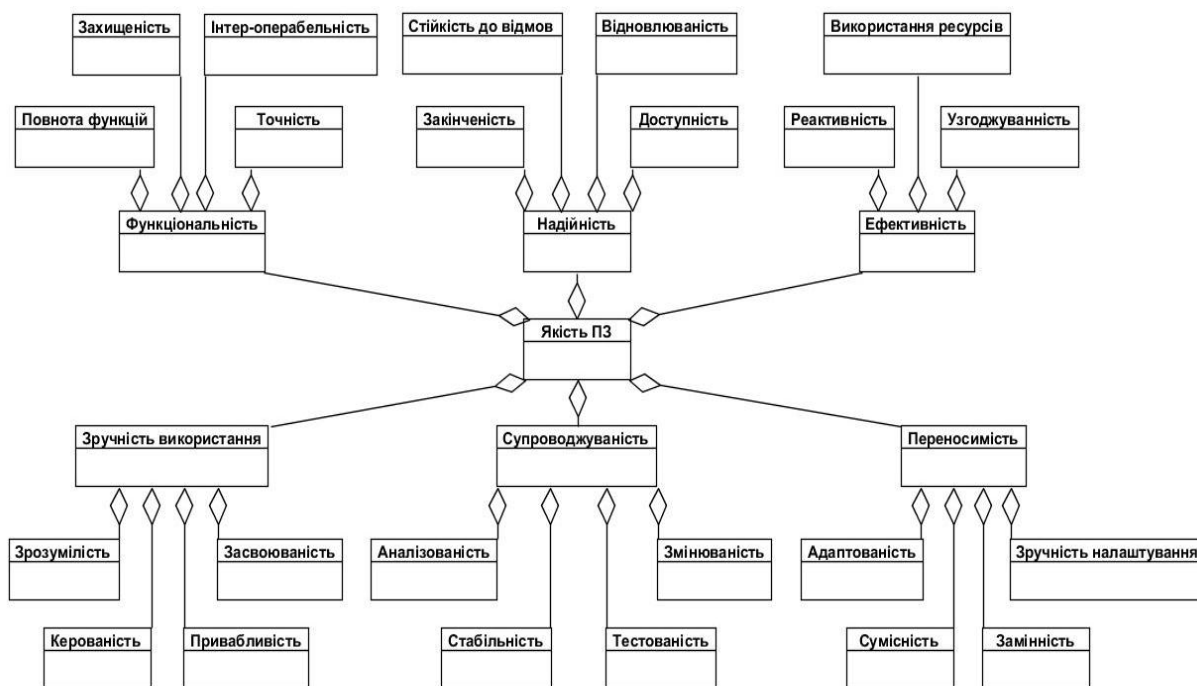


Рис. 1– Характеристики та атрибути якості

Таблиця 1 – Характеристики якості програмного забезпечення

Характеристика	Опис
Функціональність	Група властивостей ПЗ, що обумовлює його здатність виконувати визначений перелік функцій, які задовольняють потреби відповідно до призначення
Надійність	Група властивостей, що обумовлює здатність програмного забезпечення зберігати працездатність і перетворювати вихідні дані в результат за встановлений період часу, характер відмов якого є наслідком внутрішніх дефектів і умов його застосування
Зручність використання	Сукупність властивостей програмного забезпечення для передбачуваного кола користувачів, що відображають легкість його, і адаптації до умов, що змінюються, експлуатації, стабільність роботи й підготовки даних, зрозумілість результатів, зручності внесення змін у програмну документацію та у програми
Можливість супроводу	Група властивостей, що визначає зусилля, необхідні для виконання, пристосованість до діагностики відмов і наслідків внесення змін, модифікації й атестації модифікованого програмного забезпечення
Ефективність	Група властивостей, що характеризується ступенем відповідності використовуваних ресурсів середовища функціонування рівня якості (надійності) функціонування програмного забезпечення при заданих умовах застосування
Здатність до перенесення	Група властивостей програмного забезпечення, що забезпечує його пристосованість для переносу з одного середовища функціонування в інші, зусилля для переносу й адаптацію ПЗ до нового середовища

Методи тестування можна поділити на дві групи: метод ручного тестування та тестування за допомогою комп'ютера. Метод ручного тестування є, по суті, первинним виявленням помилок. Такі методи досить ефективні з точки зору знаходження помилок. Методи такого типу призначені для етапу розробки, коли програма вже закодована, але тестування на комп'ютері ще не почалося.

Методи тестування на комп'ютері мають різні стратегії тестування, до яких належать тестування

чорного, білого та сірого ящика. Дані методи, призначені для тестування не програмного комплексу в цілому, а для тестування, насамперед, програмного коду.

Методи тестування поділяються за об'єктами тестування на наступні види [1]:

- функціональне тестування (functional testing);
- тестування інтерфейсу користувача (UI testing);
- тестування локалізації (localization testing);

- тестування швидкості й надійності (load/stress/performance testing);
- тестування безпеки (security testing);
- тестування досвіду користувача (usability testing);
- тестування сумісності (compatibility testing).

Таким чином, дослідження програмних систем щодо рівня якості є важливим завданням процесу розробки. Існуючі стандарти в сфері якості програмного забезпечення визначають моделі та метрики якості, але питання щодо вибору методологічної бази тестування залишаються відкритими. В роботі розглянуто підходи до тестування зручності використання та запропоновано підхід щодо дослідження рівня зручності використання програмного забезпечення, яке розробляється.

Таблиця 2 – Атрибути функціональності

Атрибут	Опис
Повнота функцій	Властивість компонента, що показує ступінь достатності основних функцій для рішення задач відповідно до призначення програмного забезпечення
Точність	Показує ступінь досягнення правильних результатів
Інтероперабельність	Показує можливість програмного забезпечення взаємодіяти з спеціальними системами та середовищами
Захищеність	Показує здатність програмного забезпечення запобігати несанкціонованому доступу (випадковому або навмисному) до програм і даних

Таблиця 3 – Атрибути надійності

Атрибут	Опис
Стійкість до відмов	Визначає здатність ПЗ функціонувати без відмов (як програми, так і обладнання)
Відновлюваність	Здатність програми до перезапуску для повторного виконання й відновлення даних після відмов
Закінченість	Визначається величиною, зворотною до кількості відмов
Доступність	Відповідність стандартам надійності

Таблиця 4 – Атрибути зручності використання

Атрибут	Опис
Зрозумілість	Сукупність властивостей системи, що забезпечують користувачеві можливість зрозуміти, чи дійсно система може задовольнити його потреби, як вона може використовуватися для рішення визначених задач та які умови її використання
Засвоєваність	Властивості системи, що забезпечують користувачеві можливість засвоїти прийоми її використання
Керованість	Забезпечує користувачеві можливість керувати та контролювати дії системи
Привабливість	Властивості ПЗ, що визначають, чи подобається вона користувачу

Таблиця 5 – Атрибути ефективності

Атрибут	Опис
Реактивність	Показує час відгуку, обробки й виконання функцій
Використання ресурсів	Показує кількість і тривалість використовуваних ресурсів при виконанні функцій програмного забезпечення
Узгоджуваність	Показує відповідність даного атрибута заданим стандартам

Таблиця 6 – Атрибути переносимості

Атрибут	Опис
Адаптивність	Визначає зусилля, що витрачаються на адаптацію до різних середовищ
Зручність налаштування	Визначає необхідні зусилля для запуску даного ПЗ в спеціальному середовищі
Сумісність	Визначає можливість використання спеціального ПЗ в середовищі діючої системи
Замінність	Забезпечує можливість взаємодії при спільній роботі з іншими програмами з необхідною інсталяцією або адаптацією ПЗ

Таблиця 7 – Атрибути супроводжуваності

Атрибут	Опис
Аналізованість	Визначає необхідні зусилля для діагностики відмов або ідентифікації частин, які будуть модифікуватися
Змінюваність	Визначає видалення помилок у ПЗ або внесення змін для їхнього усунення, а також введення нових можливостей у ПЗ або в середовище функціонування
Стабільність	Вказує на сталість структури ПЗ й ризик її модифікації
Тестованість	Вказує на зусилля при проведенні валідації й верифікації з метою виявлення невідповідностей вимогам, а також на необхідність проведення модифікації ПЗ й сертифікації

Метою роботи є розробка моделі дослідження зручності використання в процесі тестування програмного забезпечення.

Аналіз стратегій та методів дослідження зручності використання. Для того, щоб дослідити зручність використання програмного забезпечення, необхідно визначитись з методом оцінки цієї характеристики якості. Чотири стратегії оцінювання зручності використання відображають різні фази життєвого циклу розробки програмного продукту. Життєвий цикл проходить від аналізу та дизайну до створення прототипів, конструювання та впровадження розробленої системи. Цим фазам строго відповідають стратегії оцінювання зручності використання [2]:

- віртуальне проектування (ВП) на стадії дизайну;
- моделювання програмного забезпечення (МПЗ) під час розробки прототипів;
- жорсткий огляд (ЖО) під час конструювання;

- стратегія реального світу (CPC) на стадії впровадження.

Окрім цього, стратегії також залежать від доступних ресурсів. Наприклад, коли існують реальні користувачі та реальні робочі комп'ютери, може бути використана стратегія реального світу, що включає роботу зі справжніми користувачами, розв'язання реальних завдань, використання справжнього обладнання у реальному середовищі. Зовсім протилежна стратегія необхідна, коли користувачі та комп'ютери є репрезентативними. В такому випадку можуть бути використані аналітичні методи проектування, що відображають стратегію віртуального проектування. Третя стратегія залучає реальних користувачів та репрезентативні комп'ютери. В цьому випадку програмне забезпечення розглядається у взаємодії з користувачами. Оскільки моделювання є суттєвою частиною цієї стратегії, вона називається моделюванням програмного забезпечення. Четверта стратегія залучає репрезентативних користувачів (включно з експертами, що мають критичну точку зору) та реальні комп'ютери. За такої конфігурації мова йде про стратегію жорсткого огляду. Така класифікація подана на рис. 2.

Лише при використанні певних стратегій деякі методи оцінки зручності використання підходять для

такого дослідження. Наприклад, спостереження не підходить при стратегії віртуального проектування, оскільки немає справжнього користувача, за яким можна спостерігати. З іншого боку, деякі методи є фундаментальними для певних стратегій.



Рис. 2 – Стратегії оцінювання зручності використання

Різні методи оцінювання зручності використання наведені у табл. 8 [3, 4].

У межах даного дослідження обрано стратегію реального світу, оскільки оцінювання зручності використання розробленого програмного забезпечення за допомогою реальних користувачів на реальному обладнанні дає найточніші результати. Порівняльний аналіз цих методів наведено в табл. 9.

Таблиця 8 – Методи оцінювання зручності використання

Метод	Опис
Спостереження	Особа, що займається оцінюванням зручності використання, виступає спостерігачем за користувачами під час їхньої роботи з програмним забезпеченням на комп'ютерах, помічаючи їхні успіхи, складності, вподобання, ставлення.
Опитування	Використання сукупності запитань або тверджень для отримання статистичних даних про навички, досвід, вимоги, ставлення користувачів
Інтерв'ю	Зустріч особи, що займається оцінюванням, з користувачами, проведення консультацій з метою отримання інформації про досвід роботи з програмним забезпеченням, вимоги, думки, вподобання
Емпіричний	Тестування визначених гіпотез шляхом вимірювання поведінки користувача в той час, як особа, що займається оцінюванням зручності використання, змінює параметри гіпотези
Метод груп користувачів	Використання надлишку знань та досвіду відібраної та організованої групи кінцевих користувачів
Наскрізний когнітивний аналіз	Покрокове оцінювання зручності використання когнітивним психологом з метою виявлення потенційних психологічних труднощів, що можуть виникнути у користувача з програмним забезпеченням
Евристичний	Використання програмного забезпечення або прототипу командою спеціалістів, що оцінюють зручність використання, з метою підтвердження його відповідності визнаній практиці та принципам
Оглядовий	Огляд експериментальних та емпіричних практик у відповідній літературі та встановлених стандартах
Метод моделювання	Використання моделей, таких як GOMS (Goals, Operations, Methods and Selection – цілі, операції, методи та вибір) та KLM (Keystroke Level Modelling – моделювання кількості натискань кнопок) для прогнозування зворотного зв'язку з користувачем

Таблиця 9 – Порівняння методів оцінювання зручності використання

Метод оцінювання	Стратегія	Розташування (лабораторія, робоче середовище)	Обсяг аналізованої інформації	Невідкладність отримання результатів	Нав'язливість	Вартість
1	2	3	4	5	6	7
Спостереження	CPC, МПЗ	лабораторія, РС	Великий, малий	Негайні, із затримкою	Так	Висока, низька
Опитування	CPC, МПЗ	лабораторія, РС	Великий	Із затримкою	Ні	Низька
Інтерв'ю	CPC, МПЗ	лабораторія, РС	Великий	Негайні	Так	Низька
Емпіричний	CPC, МПЗ	лабораторія	Великий, малий	Негайні, із затримкою	Так	Висока

Закінчення таблиці 9

1	2	3	4	5	6	7
Групи користувачів	СРС, МПЗ	лабораторія, РС	Великий, малий	Негайні, із затримкою	Ні	Низька
Наскрізний когнітивний аналіз	ВП	лабораторія	Малий	Негайні, із затримкою	Ні	Низька
Евристичний	ВП, ЖО	лабораторія	Великий, малий	Негайні	Ні	Низька
Оглядовий	ВП	лабораторія	Залежить від джерела	Залежить від джерела	Ні	Низька
GOMS	ВП	лабораторія	Малий	Із затримкою	Ні	Низька

Для дослідження зручності використання розробленого програмного забезпечення у межах стратегії реального світу буде використовуватися метод опитування [5, 6]. Це рішення обумовлюється низькою вартістю дослідження та ненав'язливістю даного методу. До того ж, попри затримку в отриманні результатів, можна казати про їхню точність, оскільки опитування надає можливість обробки великого обсягу статистичних даних, що підвищує надійність вимірювання [7].

Розробка методу оцінювання зручності використання на основі опитування SUMI. Для проведення дослідження зручності використання розробленого програмного забезпечення використано метод опитування користувачів Software Usability Measurement Inventory (SUMI) [8], який був розроблений в рамках проекту «Metrics for Usability Standards in Computing» (MUSiC, CECESPRIT проект номер 5429) дослідницькою групою Human Factors Research Group (HFRG).

SUMI – це рішення задачі вимірювання сприйняття користувачами зручності використання програмного забезпечення [8, 9]. Цей метод надає справедливі та надійні результати оцінювання зручності використання програмного продукту. Він дозволяє порівнювати як різні версії одного й того самого продукту, так і різні, конкуруючі програмні рішення. SUMI це об'єктивний метод оцінювання задоволення користувачів програмного забезпечення. Цей засіб складається з 50 тверджень анкети, що пройшли валідацію та верифікацію [8]. Респонденти повинні оцінити кожне твердження за трибальною шкалою (згодний, не вирішив, незгодний). Протягом створення даної анкети наведені твердження були стандартизовані як засіб для вимірювання вимог користувачів, що стосуються зручності використання програмного забезпечення. Такі вимоги наведені у Європейській директиві по мінімальним вимогам охорони здоров'я та безпеки під час роботи з дисплейними екранами (European Directive on Minimum Health and Safety Requirements for Work with Display Screen Equipment (90/270/EEC)). Також SUMI зазначається у стандарті ISO 9241 як визнаний метод дослідження зручності використання [8, 9].

Опитування вимірює такі аспекти задоволення користувачів [10]:

- ефективність;
- емоційна реакція;

- здатність допомагати;
- контроль;
- засвоєність.

Ефективність відображає відчуття користувача, чи програмне забезпечення виконує поставлені задачі швидко, ефективно, економно витрачаючи ресурси або навпаки. Емоційна реакція характеризує психологічні емоційні відчуття. В даному випадку цей аспект визначає, наскільки користувач відчуває себе розумово стимульованим, задоволеним або навпаки внаслідок роботи з програмним забезпеченням. Здатність допомагати відображає сприйняття користувача того, як програмний продукт з ним взаємодіє, допомагає усунути проблеми, що виникли. Контроль – це ступінь, з яким користувач відчуває, що він, а не програмне забезпечення, задає хід та ритм роботи. Засвоєність визначає, наскільки користувачу легко навчитися працювати з програмним забезпеченням та вивчити його нові функції.

Зазвичай користувачу потрібно 10 хвилин, щоб пройти дане опитування. Мінімальний розмір вибірки користувачів для отримання надійних результатів складає 10–12 чоловік.

Для дослідження зручності використання розробленого програмного забезпечення порівняно зручність використання розробленої системи з аналогічними. Для цього сформовано множину програмних продуктів, що порівнюються. Потім декілька користувачів за допомогою методу SUMI оцінили запропоноване програмне забезпечення. На основі індивідуальних оцінок отримано групову оцінку кожного продукту та їх порівняння.

Щоб дослідити зручність використання розробленого програмного забезпечення, використана анкета SUMI. Відповіді користувачів інтерпретовано відповідно до дихотомічної моделі Раша [8]. Тобто якщо користувач обрав твердження, що свідчить про високу зручність використання програмного забезпечення, то дане твердження для даного програмного забезпечення оцінюється в 1 бал, у протилежному випадку – у 0 балів.

Оброблялися отримані відповіді на основі моделі Раша. Отримані від різних користувачів оцінки зручності використання різних програмних продуктів зведено у загальну оцінку зручності використання кожного продукту.

Отже, структура вимірювача зручності використання показана на рис. 3.



Рис. 3 – Структура вимірювача зручності використання

На діаграмі прецедентів показані актори та можливі варіанти використання ними даного вимірювального інструменту (рис. 4).

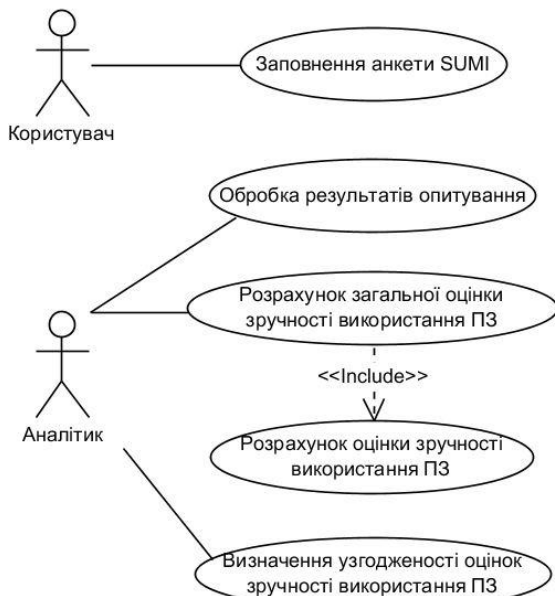


Рис. 4 – Варіанти використання вимірювача зручності використання

Для отримання індивідуальних оцінок зручності використання програмного забезпечення використовується модель Раша.

Процес отримання оцінок зручності використання визначених програмних продуктів показано на рис. 5.

Для розрахунку загальних оцінок зручності використання ПЗ можна користуватися методами обробки результатів експертного оцінювання [11]. Для цього найчастіше використовують три види середньозважених оцінок:

- середнє арифметичне зважене;
- середнє геометричне зважене;
- середнє гармонічне зважене.

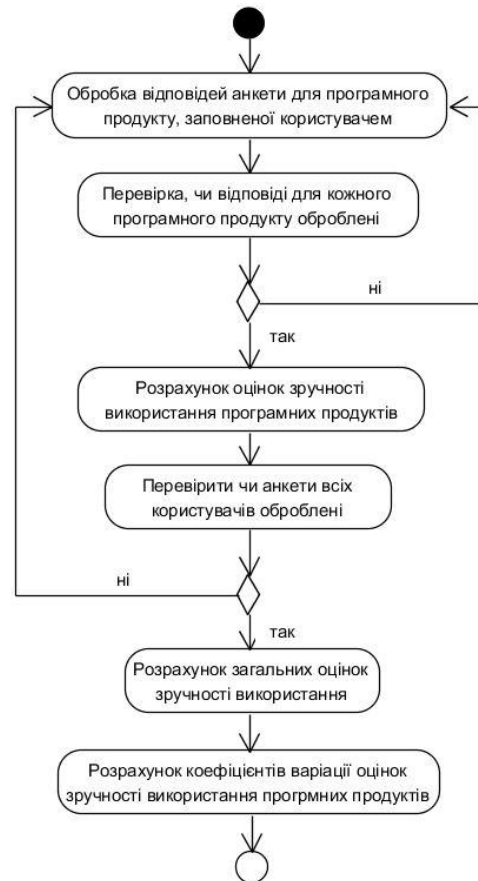


Рис. 5 – Процес оцінювання зручності використання

Середнє арифметичне зважене набору дійсних чисел x_1, \dots, x_n з ваговими коефіцієнтами w_1, \dots, w_n визначається як

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i},$$

де n – кількість оцінок, що розглядається.

Якщо сума вагових коефіцієнтів дорівнює 1, то формула має вигляд:

$$\bar{x} = \sum_{i=1}^n w_i x_i.$$

В тому випадку, якщо усі вагові коефіцієнти рівні, то середнє арифметичне зважене буде дорівнювати середньому арифметичному.

Середнє зважене геометричне визначається як:

$$\bar{x} = \left(\prod_{i=1}^n x_i^{w_i} \right)^{\frac{1}{\sum_{i=1}^n w_i}} = \exp\left(\frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i \ln x_i \right).$$

В тому випадку, якщо усі вагові коефіцієнти рівні, то середнє геометричне зважене дорівнює середньому геометричному.

Середнє зважене гармонічне визначається за формулою

$$\bar{x} = \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n \frac{w_i}{x_i}}$$

Ступінь узгодженості думок експертів відносно одиничної властивості визначається за допомогою коефіцієнта варіації v , що дорівнює

$$v = \frac{\sigma}{\bar{Q}}$$

де σ – середнє квадратичне відхилення;

\bar{Q} – середня оцінка, розрахована по одній з наведених вище формул.

На основі розрахунків коефіцієнта варіації за формулою можна зробити висновок щодо узгодженості думок експертів. Для цього можна скористатися даними, запропонованими в роботі [11]. Залежність узгодженості експертних оцінок від значення коефіцієнта варіації наведено в табл. 10.

Таблиця 10 – Визначення узгодженості експертних оцінок

Коефіцієнт варіації	Узгодженість
0,1	Висока
0,11–0,15	Вище, ніж середня
0,16–0,25	Середня
0,26–0,35	Нижче, ніж середня
0,36 та більше	Низька

Стандартне відхилення результатів експерименту визначається як [11]:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\bar{Q} - Q_i)^2 \alpha_i}{\sum_{i=1}^n \alpha_i}}$$

де n – кількість експертів, оцінки яких перевіряються на узгодженість;

α_i – ваговий коефіцієнт (наприклад, компетентність) i -го експерта;

Q_i – оцінка, надана i -м експертом;

\bar{Q} – середня оцінка.

У даній роботі пропонується брати середнє арифметичне оцінок користувачів.

Результати дослідження зручності використання програмного забезпечення. Розглянемо приклад застосування розробленого підходу. Було взято для аналізу такі програмні продукти, як GEQAS, Клієнтський модуль, Winsteps, RUMM, Facets [11, 12].

Для отримання надійних результатів було опитано 12 користувачів за допомогою анкети SUMI. Їх відповіді були інтерпретовані та оброблені на основі моделі Раша. В результаті були отримані оцінки зручності використання програмних рішень, наведені у табл. 11.

Таблиця 11 – Оцінки зручності використання програмних продуктів

№ користувача	GEQAS	KM	Winsteps	RUMM	Facets	Надійність
1	1,62	0,13	0,7	1,08	0,44	0,64
2	1,58	0,135	0,72	0,96	0,44	0,73
3	1,61	0,12	0,75	1,07	0,41	0,64
4	1,62	0,137	0,7	1,1	0,46	0,72
5	1,6	0,128	0,73	1,1	0,42	0,75
6	1,55	0,12	0,67	1,04	0,44	0,66
7	1,63	0,128	0,73	0,98	0,42	0,78
8	1,6	0,11	0,72	1,06	0,435	0,72
9	1,62	0,127	0,72	1,03	0,5	0,7
10	1,57	0,13	0,67	1,05	0,41	0,75
11	1,54	0,12	0,72	1,1	0,42	0,74
12	1,7	0,131	0,73	1,06	0,46	0,71

На основі отриманих індивідуальних оцінок розрахуємо загальні оцінки зручності використання програмних продуктів як середнє арифметичне. Також розрахуємо стандартне квадратичне відхилення та

коефіцієнт варіації для кожного програмного продукту. Результати оцінювання зручності використання наведені у табл. 12.

Таблиця 12 – Загальні оцінки зручності використання програмних продуктів

Показник	GEQAS	KM	Winsteps	RUMM	Facets
Загальна оцінка	1,6	0,13	0,71	1,05	0,44
Стандартне відхилення	0,14	0,03	0,08	0,15	0,09
Коефіцієнт варіації	0,09	0,19	0,11	0,14	0,196

На основі наведених результатів розрахунку коефіцієнтів варіації можна сказати, що для кожного програмного продукту думки користувачів характеризуються високою, вищою, ніж середня або середньою узгодженістю.

Висновки. Таким чином, результатами навантажувального тестування підтверджено стабільність роботи розробленого програмного забезпечення. Для дослідження зручності використання розробленого програмного забезпечення обрано метод опитування завдяки його низькій вартості та ненав'язливості. Зручність використання розробленого програмного компоненту оцінено за допомогою Software Usability Measurement Inventory (SUMI). Для забезпечення надійності результатів було опитано 12 користувачів, які відповіли на питання анкети SUMI. Їх відповіді були інтерпретовані та оброблені на основі моделі Раша.

Список літератури

1. ISO/IEC 9126-1:2001 Software engineering – Product quality. Part 1 : Quality model. URL : http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749 (дата звертання : 20 грудня 2017).
2. Kaner C., Falk J. Testing computer software. Thomson, 2001.
3. Fitzpatrick R. Strategies for evaluating software usability URL : <http://www.comp.dit.ie/rfitzpatrick/papers/chi99%20-strategies.pdf>. (дата звертання : 21 грудня 2017).
4. Jerz D. G. Usability Testing: What Is It? URL : <https://jerz.setonhill.edu/writing/technical-writing/usability-testing> (дата звертання : 23 грудня 2017).
5. Andreasen M. S., Nielsen H. V., Schroder S. O. Stage. What happened to remote usability testing? // Proceedings of the SIGCHI conference on Human factors in computing systems. CHI, 2007. P. 1405–1410.
6. Kapil C. M., Greenstein J. S. Synchronous remote usability testing: a new approach facilitated by virtual worlds // Proceedings of the 2011 annual conference on Human factors in computing systems. CHI, 2011. P. 2225–2234.
7. Kirakowski J. The Use of Questionnaire Methods for Usability Assessment. URL : <http://sumi.ucc.ie/sumipapp.html>. (дата звертання : 20 грудня 2017).
8. Arh T., Blazic B. A Case Study of Usability Testing – the SUMI Evaluation Approach of the Educa Next Portal. URL :

<http://www.wseas.us/e-library/transactions/information/2008/25-191.pdf> (дата звертання : 22 грудня 2017).

9. Bevan N., Curson I. Methods for Measuring Usability. URL : <http://www.nigelbevan.com/papers/meatut97.pdf>. (дата звертання : 21 грудня 2017).
10. ISO 9241-11 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11 : Guidance on usability.
11. Zhao Y., Hambleton R. Software for IRT analysis: description and features. URL : <http://www.umass.edu/remf/software/CEA-652.ZH-IRTSoftware.pdf> (дата звертання : 15 листопада 2017).
12. Sick J. Rasch analysis software programs // JALT Testing & Evaluation SIG Newsletter. 2005. № 13 (3). P. 13–16.

References (transliterated)

1. ISO/IEC 9126-1:2001 Software engineering – Product quality. Part 1 : Quality model. Available at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749 (accessed 20.12.2017).
2. Kaner C. Testing computer software. Thomson, 2001.
3. Fitzpatrick R. Strategies for evaluating software usability. Available at: <http://www.comp.dit.ie/rfitzpatrick/papers/chi99%20-strategies.pdf> (accessed 21.12.2017).
4. Jerz D. G. Usability Testing: What Is It? Available at: <https://jerz.setonhill.edu/writing/technical-writing/usability-testing> (accessed 23.12.2017).
5. Andreasen M. S., Nielsen H. V., Schroder S. O. Stage. What happened to remote usability testing? Proceedings of the SIGCHI conference on Human factors in computing systems. CHI, 2007, pp. 1405–1410.
6. Kapil C. M., Greenstein J. S. Synchronous remote usability testing: a new approach facilitated by virtual worlds. Proceedings of the 2011 annual conference on Human factors in computing systems. CHI, 2011, pp. 2225–2234.
7. Kirakowski J. The Use of Questionnaire Methods for Usability Assessment. Available at: <http://sumi.ucc.ie/sumipapp.html> (accessed 20.12.2017).
8. Arh T., Blazic B. A Case Study of Usability Testing – the SUMI Evaluation Approach of the Educa Next Portal. Available at: <http://www.wseas.us/e-library/transactions/information/2008/25-191.pdf> (accessed 22.12.2017).
9. Bevan N., Curson I. Methods for Measuring Usability. Available at: <http://www.nigelbevan.com/papers/meatut97.pdf> (accessed 21.12.2017).
10. ISO 9241-11 Ergonomic requirements for office work with visual display terminals (VDTs) Part 11 : Guidance on usability.
11. Zhao Y., Hambleton R. Software for IRT analysis: description and features. Available at: <http://www.umass.edu/remf/software/CEA-652.ZH-IRTSoftware.pdf> (accessed 15.11.2017).
12. Sick J. Rasch analysis software programs. JALT Testing & Evaluation SIG Newsletter. 2005, no. 13 (3), pp. 13–16.

Надійшла (received) 05.12.2017

Відомості про авторів / Сведения об авторах / About the Authors

Ісаков Олексій Сергійович (Исаков Алексей Сергеевич, Isakov Oleksii Serhiiovych) – Національний технічний університет «Харківський політехнічний інститут», аспірант кафедри програмної інженерії та інформаційних технологій управління; тел.: (098) 302-00-18; e-mail: isakov.ktk@gmail.com. ORCID: 0000-0002-6951-0370.

Чередніченко Ольга Юрїївна (Чередниченко Ольга Юрьевна, Cherednichenko Olga Yuriivna) – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри програмної інженерії та інформаційних технологій управління; тел.: (067) 754-79-44; e-mail: olha.cherednichenko@gmail.com. ORCID: 0000-0002-9391-5220.

Мозгін Володимир Володимирович (Мозгин Владимир Владимирович, Mozghin Volodymyr Volodymyrovych) – Національний технічний університет «Харківський політехнічний інститут», студент; тел.: (093) 267-76-33; e-mail: mozgin.vova@gmail.com. ORCID: 0000-0003-1667-1899.

Янголенко Ольга Василівна (Янголенко Ольга Васильевна, Yanholenko Olha Vasylivna) – кандидат технічних наук, Національний технічний університет «Харківський політехнічний інститут», асистент кафедри програмної інженерії та інформаційних технологій управління; тел.: (098) 438-11-14; e-mail: olga_ya26@mail.ru. ORCID: 0000-0001-7755-1255.