

ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.5 (045)

І.В. Гученко

Національний авіаційний університет

ПРОГРАМНА СИСТЕМА ОЦІНКИ ТА УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИ ГНУЧКІЙ РОЗРОБЦІ

Стаття присвячена опису програмної системи, яка реалізує розроблений автором метод ітераційної автоматизованої оцінки та управління зручністю використання програмного забезпечення при гнучкій розробці з метою врахування відгуків користувачів. Надається характеристика основних компонентів, їх призначення та особливостей програмної реалізації. Описується оцінка працездатності розробленої програмної системи, а саме: проведено емпіричне дослідження та отримані результати.

Статья посвящена описанию программной системы, реализующей разработанный автором метод итерационной автоматизированной оценки и управления удобством использования программного обеспечения при гибкой разработке с целью учета отзывов пользователей. Дается характеристика основных компонентов, их предназначения и особенностей программной реализации. Описывается оценка работоспособности разработанного программной системы, а именно: проведенное эмпирическое исследование и полученные результаты.

The article describes the software system that implements developed by the author method of iterative automated assessment and management of software usability under agile development for the purpose of taking into account users' feedback. Characteristics of basic components, their purpose and features of program implementation are given. The evaluation of efficiency of developed software is described, namely: an empirical study and its results

Ключові слова: програмне забезпечення, зручність використання, гнучка розробка, відгуки користувачів, автоматизована оцінка зручності використання, управління зручністю використання.

Вступ

Питання визначення та вимірювання зручності використання (ЗВ) програмного забезпечення (ПЗ) розглядаються в багатьох міжнародних стандартах і моделях. Проте не можна казати про узгодженість існуючих визначень, так само, як і методів вимірювання. [1] Найбільш поширеними є означення ЗВ, наведені в стандартах.

«Легкість, з якою користувач може навчитися оперувати, готувати вхідні та інтерпретувати вихідні дані системи чи компонента». [2]

«Набір властивостей, що стосуються набору робіт, необхідних для використання та індивідуальної оцінки такого використання визначеним або передбаченим колом користувачів» [3].

«Міра, в якій продукт може бути використаний певними користувачами для досягнення визначених цілей з продуктивністю, ефективністю

та задоволеністю у певному контексті використання». [4]

«Здатність програмних продуктів бути зрозумілими, вивченими, використаними та привабливими для користувача при застосуванні в заданих умовах». [5]

Останні дослідження в цій галузі викладено в праці [6], де описано об'єднану модель QUIM (Quality in Use Integrated Measurement), яка включає 10 факторів (атрибутів) зручності використання/якості у використанні, що розкладаються на 26 субфакторів (показників) і 128 метрик. Модель враховує наступні атрибути ЗВ: економічність, ефективність, продуктивність, задоволеність, можливість навчання, безпека, довіра, доступність, універсальність і корисність.

Особливої уваги заслуговує питання забезпечення ЗВ при гнучких методах розробки, які стають все більш популярними. Ітеративність

та невибагливість згаданих методів дозволяє в них інтегрувати методи оцінки зручності використання через зосередженість на користувачах та безперервне тестування.

Малодослідженими при гнучкій розробці залишаються задачі автоматизованої оцінки поточного та забезпечення бажаного рівня зручності використання програмної системи на основі оцінок користувачів.

Дослідження автора присвячені розробленню методу ітераційної автоматизованої оцінки та управління зручністю використання програмного забезпечення при гнучкій розробці на основі відгуків користувачів.[7-10] Вказаний метод складається з наступних кроків:

Побудова експертами ієрархічної моделі атрибутів, показників та метрик ЗВ. Даний етап передбачає розробку метрик структурним методом зверху-вниз (top – down) [11] для забезпечення цілісності та повноти і містить наступні рівні:

1. Верхній рівень ієрархії: формування вимог до зручності використання; вибір атрибутів, встановлення пріоритетів та зв'язку з вимогами; встановлення за допомогою менеджерів (або замовника) допустимих коридорів для числових значень атрибутів;

2. Середній рівень ієрархії: виконання декомпозиції атрибутів ЗВ у показники, які є обчислюваними характеристиками;

3. Нижній рівень ієрархії: декомпозиція показників в метрики, які можуть бути безпосередньо оцінені в числовій формі користувачами при використанні програмного продукту. На їх основі обчислюються поточні значення показників та атрибутів верхніх рівнів ієрархії.

Формування числових значень метрик ЗВ на підставі користувацьких оцінок. Необхідно проведення опитування користувачів, які дають числову оцінку по відповідних метриках.

Отримання єдиної оцінки ЗВ програмного забезпечення.

Кожному атрибуту (показнику, метриці) в рамках ієрархічної моделі експертами предметної області приписується числова величина – «вага». Це дозволяє, за умови узгодженості експертних ранжувань (виконання умови перевіряється розрахунком коефіцієнта конкордації та застосуванням кластерного аналізу), відповідно до ієрархічної моделі зводити окремі значення критеріїв ЗВ в єдину числову оцінку.

Побудова математичної моделі управління зручністю використання. Побудована математична модель оцінки ЗВ доповнюється функцією трудомісткості зміни показників ЗВ, тим самим, отримується модель забезпечення необхідного її рівня оптимальним чином. Отже,

задача зводиться до оптимізаційної задачі при мінімізації трудомісткості робіт для забезпечення заданого рівня ЗВ.

Формування оптимальної стратегії управління ЗВ. У випадку узгодженості оцінок користувачів можлива побудова оптимальної стратегії покращення рівня ЗВ за рахунок підстановки значень показників в модель забезпечення необхідного рівня ЗВ та відшукування оптимального рішення. Для оптимізації чисельного рішення та врахування впливу змінюваних показників на значення ЗВ пропонується встановити наявність та форму зв'язку між парами розглянутих показників. Рішенням буде набір показників, які потребують покращення (з урахуванням величини зміни кожного показника). Для визначення ключових напрямків покращення, кількість показників, що потребують його, штучно обмежується.

Впровадження обраної стратегії, контроль виконання та коригування моделі.

Суттєвою відмінністю даного методу від існуючих методів оцінки ЗВ при гнучкій розробці є те, що він передбачає автоматизовану побудову стратегії забезпечення заданого рівня ЗВ на наступній ітерації оптимальним чином на основі математичних моделей оцінки та управління ЗВ, орієнтованих на використання оцінок користувачів.

Для реалізації методу створено програмну систему, яка забезпечує збір інформації про використання програмних продуктів у вигляді оцінок користувачів та суджень експертів з наступною її обробкою. Мета – визначення ключових напрямків покращення зручності використання досліджуваного програмного забезпечення.

Аналіз досліджень і публікацій

Оцінка зручності використання може бути досить дорогою, зважаючи на час та людські ресурси, тому автоматизація – зниження витрат. Автоматизована оцінка ЗВ ґрунтується на розробці методів, що забезпечують прискорення обрахунків, ширшу аудиторію для тестування зручності використання, та створенні засобів, які мають вбудовані аналітичні можливості. У роботі [12] досліджено стан автоматизації при оцінці ЗВ та наведено класифікацію існуючих методів за наступними ознаками:

– *Клас методу*: визначає тип виконуваної оцінки на вищому рівні (наприклад, тестування або симуляція);

– *Тип методу*: визначає конкретний спосіб виконання оцінки всередині класу методу, наприклад, аналіз системного журналу (клас

тестування) або моделювання мереж Петрі (клас симуляції);

– *Тип автоматизації*: визначає ступінь автоматизації оцінки (наприклад, фіксація або аналіз);

– *Рівень зусиль*: визначає кількість зусиль, необхідних для застосування методу (наприклад, розробка моделі або використання інтерфейсу).

На рис.1 представлена вищезгадана класифікація, а також показано місце розробленого методу серед інших.

Методи оцінки згідно класифікації [12] можуть належати до наступних класів:

– *Тестування (Т)*: спостереження за виконанням користувачами завдань або віддалене тестування; включає підходи, які дають кількісну оцінку метрик на основі зібраних даних щодо використання;

– *Інспекція (І)*: використання ряду критеріїв або отриманих дослідним шляхом твер-

джень для виявлення потенційних проблем ЗВ. Передбачається робота експертів, зокрема перевірка відповідності продукту вимогам стандартів;

– *Опитування (О)*: надання користувачами оцінки ЗВ шляхом інтерв'ю, анкет тощо. Також сюди відносять спостереження за використанням ПЗ в середовищі користувача;

– *Аналітичне моделювання (АМ)*: використання моделей користувачів та інтерфейсів для прогнозування;

– *Симуляція (С)*: застосування моделей користувачів та інтерфейсів для імітації взаємодії користувача з ПЗ та створення звітів щодо результатів такої взаємодії.

Усередині кожного класу існує багато методів оцінки ЗВ. Пов'язані між собою методи об'єднуються в типи. Характеристика останніх не наводиться в даній роботі через їх чисельність.



Рис.1. Підхід до класифікації методів оцінки зручності використання

Методи оцінки ЗВ можуть реалізовуватись з використанням одного з наступних типів автоматизації [12, 13]:

– *Неавтоматизована оцінка (Н)*: спеціаліст з оцінки ЗВ самостійно виконує всі кроки методу;

– Фіксація (Ф): використання програмних пристроїв для запису відповідної інформації щодо користувача та системи (наприклад, візуальних даних, мовної діяльності, дій з клавіатурою);

– Аналіз (А): автоматизоване визначення потенційних проблем ЗВ;

– Критика (К): виявлення недоліків та пропонування покращень шляхом автоматизованого аналізу.

Залежно від зусиль, необхідних для застосування методу, виділяють наступні рівні (наведені в порядку зростання трудовитрат) [12]:

– Мінімальний: не потребує використання інтерфейсу або моделювання;

– Розробка моделі: вимагає розробки моделі користувацького інтерфейсу і/або моделі користувача;

– Неформальне використання: потребує виконання довільно обраних завдань;

– Формальне використання: вимагає виконання спеціально обраних завдань.

У [12] наведено огляд більшості існуючих методів оцінки ЗВ та статистику, яка показує, що автоматизація використовується недостатньо. Методи без автоматизованої підтримки складають 67%, відповідно, методи з автоматизацією – 33 %. Із цих 33 % фіксація складає 13 %, аналіз – 18 %, а методи автоматизованої критики – усього 2 %. Останні належать класу методів інспекції.

Аналіз джерел показав, що переважна більшість методів стосується, як правило, оцінки web-інтерфейсів та графічних інтерфейсів користувача. Але дослідження, проведені автором, свідчать, що поняття зручності використання програмного забезпечення є більш багатогранним.[8] У [14] наводяться результати web-опитування щодо методів оцінки ЗВ при гнучкій розробці, основою якого стали кількісні та якісні методи дослідження. Подамо найбільш популярні методи оцінки ЗВ (рис.2), використавши класифікацію [12], у вигляді *Назва/тип методу {англомова назва} (клас методу, тип автоматизації)*:

– низькорівневе прототипування {Low-Fidelity Prototyping} (I, H);

– концептуальне проектування {Conceptual Design} (O, H);

– спостереження за користувачами {Observational Studies of Users} (O, H);

– експертна оцінка {Usability Expert Evaluations} (I, H);

– польові дослідження {Field Studies} (O, H);

– створення уявних користувачів {Personas} (I, H);

– швидке ітеративне тестування {Rapid Iterative Testing} (T, H);

– лабораторне тестування {Laboratory Usability Testing} (T, H);

– аналіз потреб {Needs Analysis} (I, H);

– цільове проектування {Goal-Directed Design} (I, H);

– віддалене тестування {Remote Usability Testing} (T, H);

– концептуальне опитування {Conceptual Inquiry} (O, H).

Пункт «Інші» містить наступні неавтоматизовані методи: контекстне опитування, неформальне дослідження ЗВ, залучення до проектування інтерфейсів з використанням специфікацій, високорівневе прототипування і опитування на основі моделей (model-driven inquiry).

Як видно з рис.2, автоматизована оцінка ЗВ при гнучкій розробці використовується недостатньо. Здебільшого це автоматизація приймального тестування, статичного аналізу коду, створення та віддалена оцінка низькорівневих прототипів.

Таким чином, у класі методів оцінки ЗВ «Тестування» (особливо тих, які використовуються при гнучкій розробці ПЗ) недостатньо методів, які відносяться до типу автоматизації «Критика». Наведена статистика обґрунтовує доцільність створення методу автоматизованої оцінки та управління ЗВ програмного забезпечення при гнучкій розробці.

Серед відомих програмних засобів, які автоматизують різні аспекти методів оцінки зручності використання програмного забезпечення, можна виділити наступні.

Обробник на основі методу GOMS (Goals, Operators, Methods, Selection rules).[15] Метод належить до класу «Аналітичне моделювання», тип автоматизації – «Аналіз». Для його реалізації було розроблено процесор, який обробляє інформацію в тому вигляді, в якому її сприймають люди.

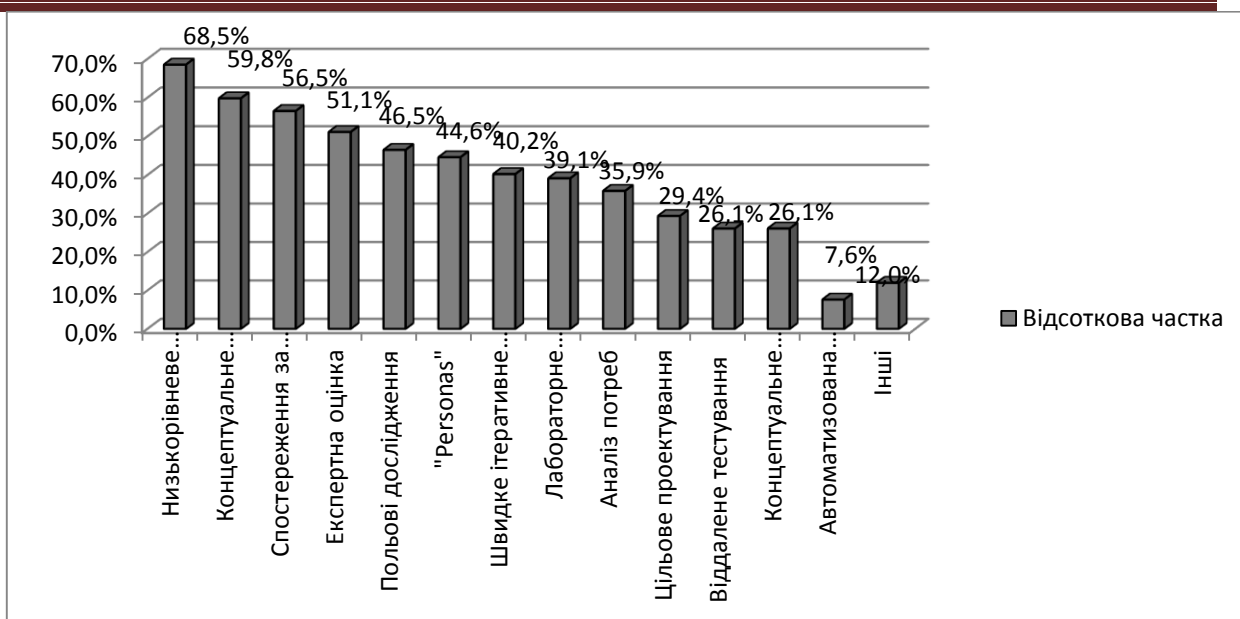


Рис.2. Використання методів оцінки ЗВ при гнучкій розробці

Відповідна модель включає можливості короткотермінової та довготермінової пам'яті, візуального сприйняття та сприйняття на слух. Також включено час, необхідний на когнітивну (пізнавальну) обробку, та швидкість обробки. Це дозволяє дослідникам людино-комп'ютерної взаємодії оцінювати проекти інтерфейсів, отримані з моделі, спираючись на прогноз щодо продуктивності. Модель GOMS складається з цілей, операторів, методів і правил вибору. Метод – це набір дій або операторів, за допомогою яких можна досягти цілі. Модель GOMS може використовуватися лише для оцінки ефективності процедурного аспекту ЗВ.

EPIC системи (Executive-Process/ Interactive Control).[16] Реалізують метод, який відноситься до класу «Симуляція», тип автоматизації – «Аналіз». Системи імітують сприйняття інформації та швидкість її обробки подібно до того, як це здійснюється мозком людини; почали використовуватися для вивчення користувачів, які працюють з декількома завданнями одночасно. Наприклад, використання навігаційної системи під час водіння автомобіля. Застосування EPIC систем включає в себе написання правил щодо використання інтерфейсу і створення середовища для імітації поведінки інтерфейсу користувача.

Інструмент AIDE (Semi-Automated Interface Designer and Evaluator).[17] Підтримує метод класу «Інспекція», тип автоматизації – «Аналіз». Представляє собою програмний інструмент для оцінки статичних сторінок HTML у відповідності з набором визначених керівних принципів web-дизайну, таких як розміщення і

вирівнювання елементів екрану (наприклад, текст, кнопки або посилання). Інструмент AIDE також може генерувати альтернативні розкладки інтерфейсу і оцінювати деякі аспекти дизайну. Дизайн оцінюється, враховуючи чутливі і нечутливі до завдань метрики. За допомогою чутливих метрик можна отримати інформацію про завдання для процесу розробки. Це може гарантувати, що завдання користувача є орієнтиром для семантики дизайну інтерфейсу. Нечутливі до завдань метрики, як правило, спираються на принципи графічного дизайну і пов'язані з естетичністю інтерфейсу. Всього інструмент AIDE може вимірювати п'ять метрик зручності використання: ефективність, вирівнювання, горизонтальний баланс, вертикальний баланс і конструкції, визначені дизайном (наприклад, позиціонування елементів).

QUIM Editor (Quality in Use Integrated Measurement Editor).[6] Підтримує метод класу «Інспекція», тип автоматизації – «Аналіз». Відповідна модель QUIM включає 10 факторів (атрибутів) зручності використання/якості у використанні, що розкладаються на 26 субфакторів (показників) і 128 метрик. Модель враховує наступні атрибути: економічність, ефективність, продуктивність, задоволеність, можливість навчання, безпека, довіра, доступність, універсальність і корисність. Відповідний інструмент розроблений для практиків, які не обов'язково мають бути експертами зі ЗВ. Редактор підтримує перегляд та вивчення відношень між множинами атрибутів, показників, метрик та даних; містить інформацію про збір даних, необхідних для різноманітних метрик; надає можливість створення та збереження

репозиторію планів з вимірювання ЗВ для різних комбінацій атрибутів, показників та метрик; містить декілька майстрів для допомоги розробникам або тестувальникам при вимірюванні ефективності систем з особливими вимогами до безпеки або доступності Web-застосувань.

DRUM (Diagnostic Recorder for Usability Measurement).[18] Реалізує метод класу «Тестування», тип автоматизації – «Аналіз». Представляє собою програмний інструмент для аналізу оцінок, отриманих від користувачів, і представлення цих даних відповідній стороні, наприклад, інженеру зі зручності використання. Цей інструмент обчислює метрики ЗВ, пов'язані з виконанням програми, в тому числі:

- Час завдання або загальний час, необхідний для виконання кожного завдання.

- Проблеми, допомога і час пошуку, які стосуються кількості часу, що користувачі витрачають, вирішуючи проблеми пошуку або непродуктивного блукання системою.

- Ефективність, яка залежить від кількості та якості виконаного завдання, а також від досягнення користувачем цілей при роботі з системою.

- Економічність, яка пов'язує ефективність з часом виконання завдання і, таким чином, рівень виконання завдання.

- Відносна ефективність, яка показує, як ефективно завдання виконується звичайним користувачем в порівнянні з досвідченим користувачем в тій же системі або з тим же завданням в іншій системі.

- Продуктивний період, або відношення часу завдання, що не проводився у пошуку допомоги, до загального часу виконання завдань (тобто кількість корисного часу роботи).

Серед інструментів, які підтримують клас методів «Тестування» та тип автоматизації «Аналіз» також широко відомі AMME (Automatic Mental Model Evaluator), MIKE (Menu Interaction Kontrol Environment) UIMS, MRP (Maximum Repeating Pattern), USINE (User Interface Evaluator) та ін.[12] Усі вони отримують дані шляхом аналізу системного журналу та виконують підрахунок окремих метрик або порівнюють моделі виконання завдань з реальними діями користувача.

Існуючі інструменти не забезпечують реалізацію розробленого методу ітераційної автоматизованої (тип автоматизації – «Критика») оцінки та управління зручністю використання програмного забезпечення при гнучкій розробці на основі відгуків користувачів. Тому пос-

тає задача створення відповідної програмної системи.

Функціональні вимоги до програмної системи та особливості архітектури

Програмна система має забезпечувати оцінку та управління ЗВ програмного забезпечення при гнучкій розробці шляхом реалізації функціональних вимог, модель яких представлено за допомогою діаграми станів [19] на рис.3.

Вимога завантаження результатів опитування в БД забезпечується наступними функціями системи:

- авторизація експерта;
- вибір шаблону форми загрузки даних;
- заповнення шаблону даними з БД;
- пересилання заповненого шаблону анкети користувачу/експерту;
- внесення даних опитування в БД;
- сповіщення користувача/експерта про успішність внесення анкетних даних.

Вимога виконання первинного статистичного аналізу даних реалізується за допомогою функцій системи:

- імпорт з БД оцінок (метрик) та рангів метрик, показників, атрибутів ЗВ;
- розрахунок математичного сподівання, медіани, дисперсії, коефіцієнтів ексцесу та асиметрії;
- побудова двовимірної гістограми розподілу значень оцінок та рангів.

Вимога оцінки узгодженості даних опитування [9] забезпечується функціями системи:

- розрахунок коефіцієнта конкордації окремодля користувачів та експертів;
- виконання кластерного аналізу на даних опитування експертів.

Вимога оцінки поточного рівня ЗВ виконується за допомогою функцій:

- введення підсумкових рангів метрик, показників, атрибутів ЗВ;
- розрахунок вагових коефіцієнтів метрик, показників, атрибутів ЗВ;
- розрахунок поточного рівня ЗВ.

Вимога оптимізації чисельного рішення моделі реалізується через наступні функції:

- розрахунок математичного сподівання, медіани, дисперсії, коефіцієнтів ексцесу та асиметрії для значень показників ЗВ;
- зміна закону розподілу згідно обраного математичного перетворення;
- розрахунок коефіцієнтів кореляції для всіх пар показників;
- вибір виду регресії для обраних пар показників;

- розрахунок коефіцієнту детермінації та рівняння регресії;
- побудова графіку лінії регресії.
- Вимога розрахунку варіантів підвищення рівня ЗВ забезпечується функціями системи:
 - введення формули залежності між показниками;
 - введення значення бажаного рівня ЗВ;
 - введення кроку перебору значень показників;
 - обчислення наборів показників, які задовольняють умовам;
 - побудова тривимірного точкового графіка, по осях: сума змін показників в розглядуваному варіанті, максимальна вага показника, початкове значення показника з максимальною вагою;
 - при виборі точки на графіку виводиться інформація щодо назви показників, їх початкових значеннях та необхідної величини зміни.

Для реалізації функціональних вимог пропонується використати гібридну архітектуру (рис.4) програмного забезпечення, яка поєднує в собі елементи клієнт-серверної архітектури [20] та архітектури, орієнтованої на БД [21]. Такий вибір обумовлений необхідністю централізованого збору даних (web-сайт) та їх довготривалого зберігання (БД).

Пропонується застосувати модель «товстого клієнта» для обробки даних на стороні клієнта-застосування, оскільки обробка є досить ресурсоємною та інтерактивною.

Система складається з трьох частин:

1. Web-сервіс, що забезпечує збір даних.
2. База даних, яка зберігає зібрану інформацію та налаштування системи.
3. Аналізатор, який забезпечує управління системою, аналіз даних та виведення результатів.

Web-сервіс. При створенні Web-сервісу використовувалась технологія Java Server Pages (JSP). Відповідні сервлети, у які перетворюються JSP документи перед використанням, поміщені у Web-контейнер Apache Tomcat [22], що забезпечує обмін даними між сервлетом та користувачами, виконує функції створення програмного середовища для функціонування сервлета, ідентифікації та авторизації експертів, організації сесії для кожного з них. Вибір вказаного контейнера пояснюється відкритістю його коду, невеликим розміром проекту і легкістю установки та налаштування. Статичні вихідні дані JSP сторінки оформлені в форматі HTML. Архітектура Web-сервісу є найпростішим однорівневим рішенням, коли JSP сторінка одночасно містить контролер

обробки запитів, шаблон представлення і Java-компонент, який здійснює завантаження даних з сервера БД.

База даних. Як було зазначено, Web-сервіс забезпечує ефективний і надійний збір даних, їх перевірку і відправку на сервер. На сервері дані зберігаються в базі, що управляється MySQL [23]. Вибір даної БД обумовлений особливостями розв'язуваної задачі оцінки та управління ЗВ програмного забезпечення: порівняно малий обсяг збережених даних і невелике число з'єднань таблиць при виконанні запитів дають можливість використовувати цю ефективну й економічну щодо ресурсів систему.

Аналізатор. Реалізований у вигляді віконної інтерактивної системи, яка забезпечує налаштування всієї системи, вибірку даних з БД, їх аналіз в напів-автоматичному режимі і виведення результатів. При розробці застосовувалися мови Java і C++. Вибір Java визначається високою ефективністю реалізації графічних користувацьких інтерфейсів і підсистеми доступу до БД. C++ використовується, оскільки, за інших рівних умов, забезпечує вищу швидкодію обчислювальних алгоритмів. Міжмовна взаємодія досягається за допомогою технології JNA. Фізичні компоненти аналізатора зображено на рис.5.

Можна виділити наступні підсистеми аналізатора:

- підсистема завантаження та первинного аналізу даних опитування,
- підсистема оцінки узгодженості відповідей користувачів/експертів та розрахунку поточного рівня ЗВ,
- підсистема побудови зв'язків всередині математичної моделі оцінки та управління ЗВ,
- підсистема розрахунку варіантів досягнення заданого рівня ЗВ.

Застосування програмної системи

Одним із важливих кроків методу оцінки та управління ЗВ на основі відгуків користувачів при гнучкій розробці є побудова моделі оцінки ЗВ. У якості формальної моделі застосовано ієрархічну модель, засновану на інтегрованій оцінці зважених показників ЗВ програмного забезпечення [10]. Для формування поточних значень атрибутів, показників та метрик використовується опитування користувачів, які дають числову оцінку по кожній метриці, а також ранжування експертів для визначення ваги критеріїв. Таким чином, вкрай важливим є визначення найбільш повного переліку критеріїв ЗВ,

яким можуть дати оцінку користувачі та виставити ранги експерти.

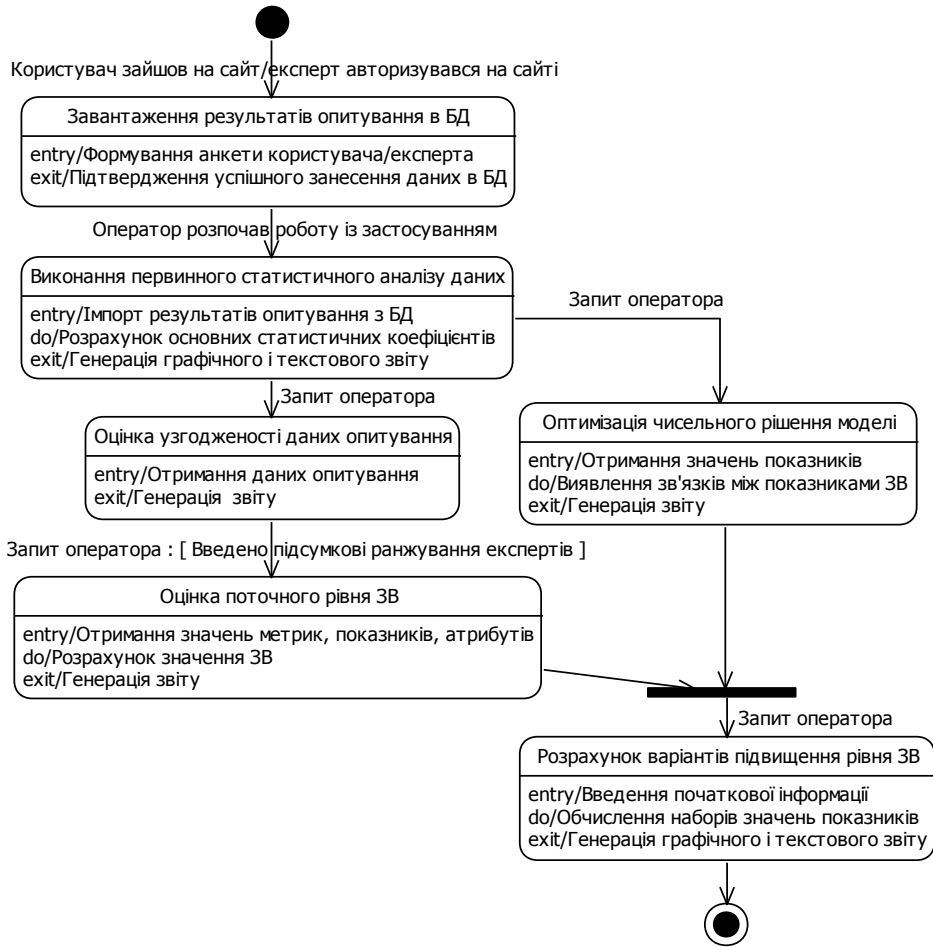


Рис.3. Модель функціональних вимог до програмної систем

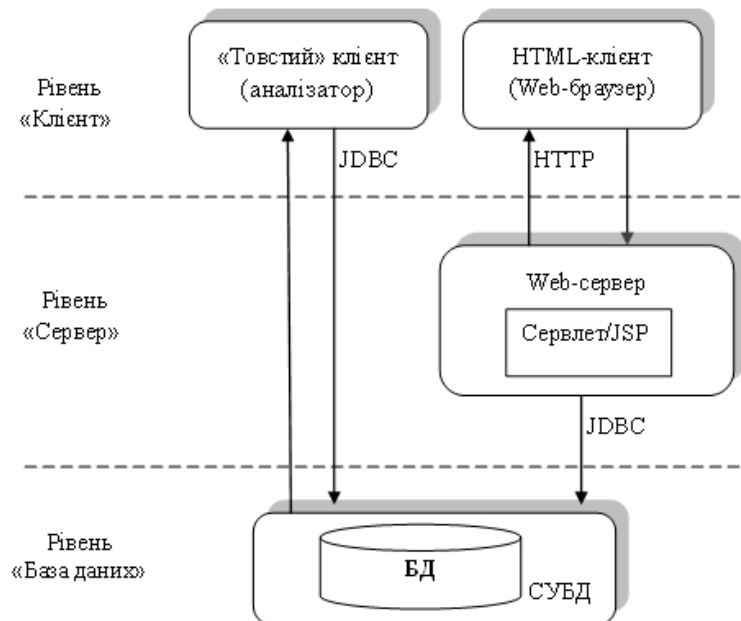


Рис.4. Архітектура програмної системи

Список атрибутів та показників ЗВ ПЗ наводиться автором в роботі [8]. Загаданий перелік має уточнюватися експертами для кожного конкретного програмного забезпечення.

Досліджуване ПЗ. У рамках попередніх досліджень було запропоновано метод оцінки зручності використання програмного забезпечення, який передбачає оцінку поточного рівня ЗВ на кожній ітерації в процесі гнучкої/ітераційної розробки та формування оптимальної стратегії досягнення бажаного рівня (який задається розробником на початку). Для оцінки методу та створеної програмної системи було обрано web-браузер Chromium з відкритим вихідним кодом, на основі якого створюється відомий широкому колу користувачів

браузер Google Chrome. Процес розробки Chromium є ітеративним з використанням системи «rapid release». Замість циклу розробки програмного забезпечення, у якому функції відточуються в кожному релізі і можуть його затримувати, Chrome випускає новий реліз будь-що кожні 6 тижнів. Якщо нові функції не готові, вони чекають наступного релізу. При цьому версія розвивається по трьох напрямках одночасно: розробника (dev), бета (beta) і стабільному (stable). Робочі релізи датуються з вересня 2008 року. Зазначені особливості Chromium дають змогу застосувати розроблені метод і засіб оцінки та управління ЗВ.

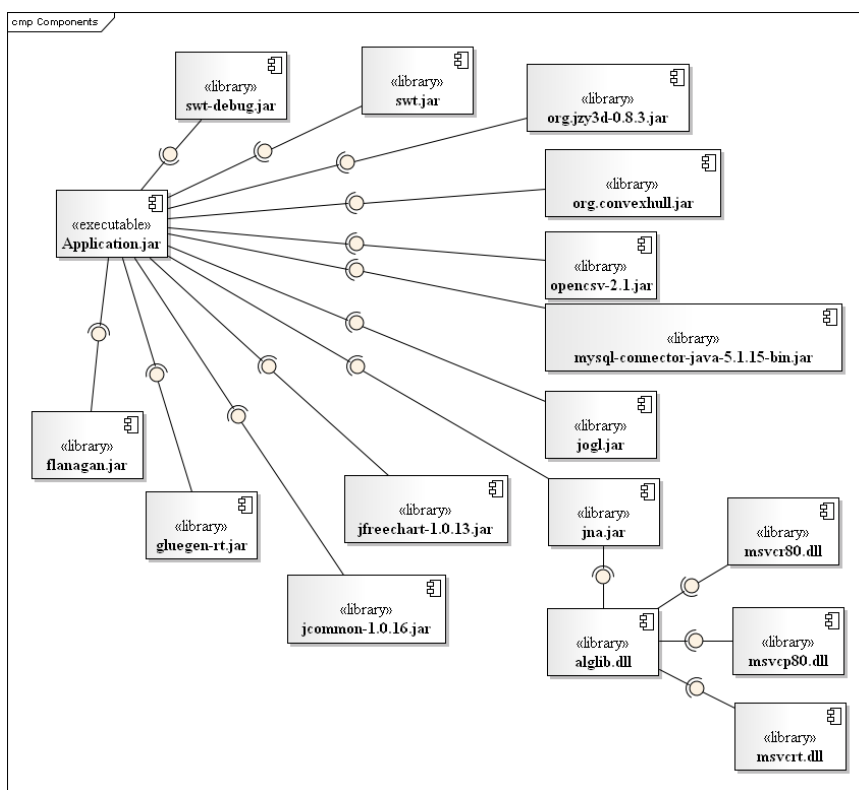


Рис.5. Фізичні компоненти аналізатора

Експериментальна оцінка та варіанти управління ЗВ. У дослідженні використовувався відомий метод емпіричної інженерії програмного забезпечення – анкетування (survey) користувачів та експертів.[24] У ролі користувачів виступила група з 52 студентів четвертого та п'ятого курсів факультету комп'ютерних наук Національного авіаційного університету. Їм було запропоновано дати кількісну оцінку (від 1 до 10 балів) метрик, обраних для аналізу досліджуваного ПЗ. Ранжування користувачами показників, атрибутів та метрик ЗВ передбачено в програмній системі оцінки та управ-

ління ЗВ, але не є обов'язковим, оскільки для визначення ваги критеріїв, як правило, використовують ранги експертів. Зауважимо, що в рамках дослідження не ставилася задача опитування кількості студентів достатньої для екстраполяції на всю генеральну сукупність з високою точністю. Це пояснюється тим, що основною метою є не виведення будь-яких статистичних закономірностей, а перевірка роботи методу оцінки та управління ЗВ програмного забезпечення. Кількість опитаних дає змогу узагальнювати отримані результати з довірчою ймовірністю 85% та довірчим інтервалом (по-

хибкою) в 10% за умови нормальності розподілу. Результати первинної обробки даних опитування показали, що розподіли оцінок в переважній більшості випадків не мають викидів та задовольняють подальшому аналізу. Нижче подано коротку характеристику деяких отриманих метрик ЗВ.

Серед запропонованих для оцінки 67 метрик ЗВ більшість розподілів характеризується ненормальним законом, значенням медіани $Me=8$ та дисперсією $0,37 < \sigma^2 < 2,8$, а отже, стандартним відхиленням $0,61 < \sigma < 1,67$. Таким чином похибка репрезентативності (середнього значення) наступна:

$$0,085 < m = \frac{\sigma}{\sqrt{n}} < 0,232,$$

де $n=52$ – кількість елементів вибірки.

Згідно заявленим розробниками Chromium прагненням створити найбезпечніший браузер, вони приділяють багато уваги новим функціям із захисту, проте, як можна побачити з розподілу оцінок метрики «Захист» (рис.6), майже 29% (15 опитуваних) виставили оцінки менше 8 балів, тобто вважають рівень здатності перешкоджати несанкціонованому доступу недостатнім.

Відсоткові співвідношення виставлених оцінок по інших окремих метриках представлено круговими діаграмами на рис.7. Час завантаження сторінок був оцінений більшістю опитуваних вище 8 балів (рис.7а),

$Me=9$. Дані результати відповідають дослідженням компанії Gomez, згідно якого швидше всього завантажуються сайти у користувачів Chrome 12 (порівняно з Firefox, Safari та Internet Explorer).[25] Середня швидкість завантаження сторінки на Chrome 12 складала 3,433 секунди. Трохи нижче опитуваними було оцінено час виконання завдання (рис.7б) та частоти виникнення помилок (рис.7в), $Me=8$.

Для виявлення останніх Google веде політику добровільного бета-тестування Chromium, а з 2010 року виплачує грошові винагороди. Стосовно користувацьких опцій (рис.7г) 46 % опитуваних оцінили на 9 з 10 балів число функцій, які можуть бути налаштовані згідно їх вподобання.

Інтерфейс за результатами дослідження є візуально простим (рис.7г), а відсоток екрану,

використовуваний для представлення інформації (рис.7д), 36 % студентів оцінили на 7, а 37 % - на 10 балів. Загалом, більшість метрик, за якими оцінювалася ЗВ web-браузеру Chromium 12, отримали високі бали.

Винятком є метрики «Конфіденційність», «Страховання» та «Безпека користувача» (рис.7е-ж).

Видно, що надійність захисту особистої інформації від сторонніх, більшість опитуваних – 44 % – оцінили на 8, але 39 % виставили нижчі бали.

Результати також вказують на недостатню впевненість в тому, що розробник Chromium 12 нестиме відповідальність у випадку шахрайського використання особистої інформації користувачів. 42 % опитуваних оцінили безпеку стану при користуванні web-браузером на 3 бали.

Останні дані відображають загальну для програмного та апаратного забезпечення вад – можливе погіршення фізичного стану, наприклад, «тунельний» ефект, почервоніння очей тощо.

Таким чином, первинний аналіз даних опитування показав досить високу оцінку користувачами більшості метрик ЗВ web-браузеру Chromium 12.

Наступним результатом дослідження було отримання експертних ранжувань для подальшого розрахунку ваги метрик, показників та атрибутів ЗВ.

У якості експертів виступили 5 фахівців предметної області. Коефіцієнт конкордації [9] для виставлених рангів атрибутів становить 0,88 та є значимим, що свідчить про сильну ступінь узгодженості за шкалою Чеддока.

Маючи оцінки користувачів та вагу критеріїв ЗВ, можна розрахувати [8] поточний рівень ЗВ Q web-браузеру Chromium 12. Згідно отриманих даних $Q=0,81$.

Перед вирішенням задачі забезпечення бажаного рівня ЗВ було виконано, використовуючи підсистему побудови зв'язків всередині математичної моделі оцінки та управління ЗВ, зниження розмірності простору перебору за допомогою усунення залежних показників шляхом виявлення емпіричних залежностей (рис.8).

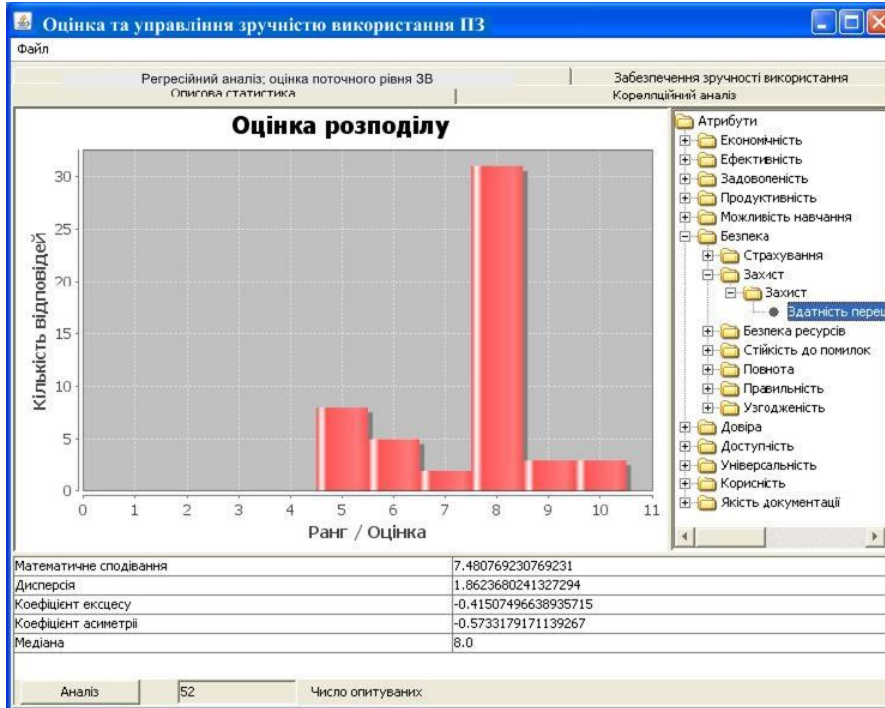


Рис.6. Розподіл оцінок метрики «Захист» web-браузеру Chromium 12

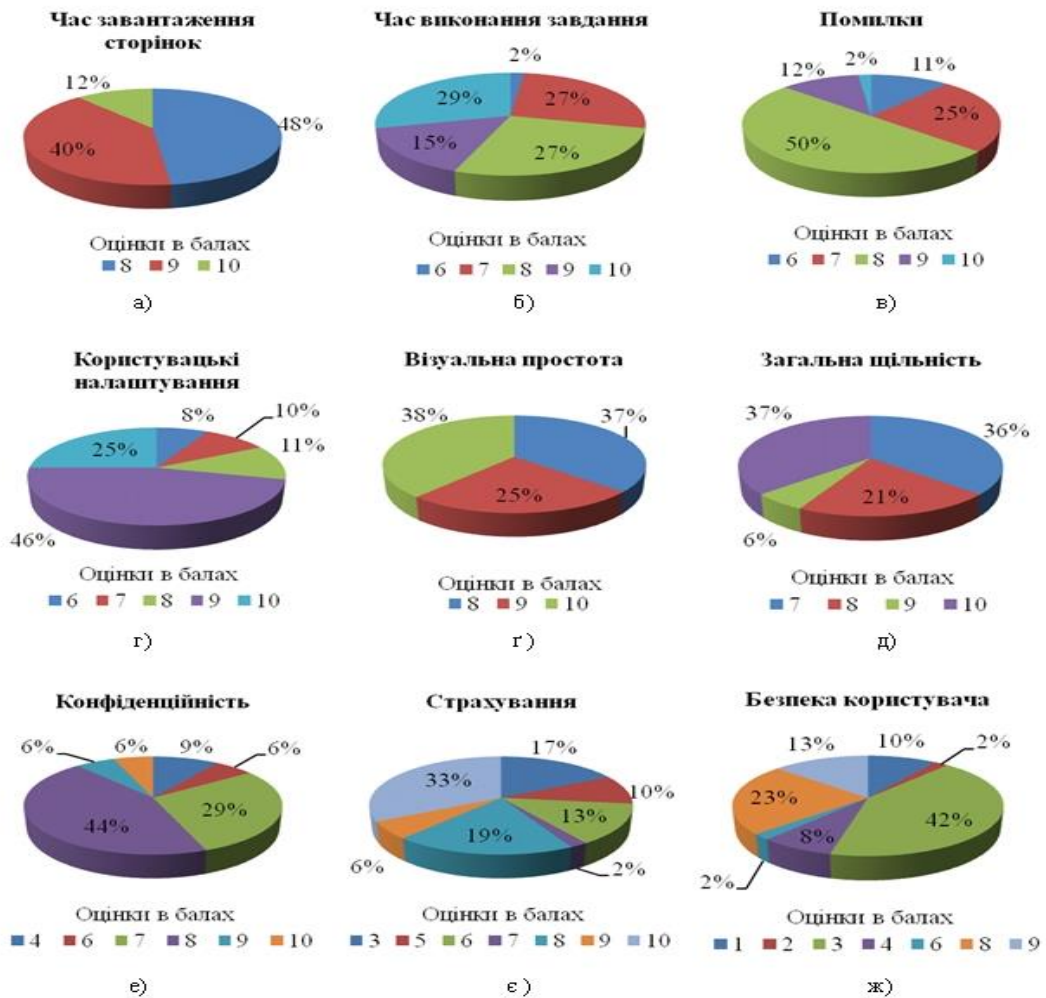


Рис.7. Оцінка web-браузеру Chromium 12 користувачами

Тіснота зв'язку між показниками визначалася за шкалою Чеддока: до уваги бралися лише ті пари показників, що мають сильний зв'язок (табл.1). Визначені зв'язки дозволяють оцінити вплив зміни одного з показників на інший. Результати дослідження показують, що всі залежності є прямо пропорційними, тобто при збільшенні (зменшенні) незалежного показника, значення залежного також збільшується (зменшується).

У рамках вирішення задачі управління ЗВ web-браузеру Chromium 12 виконується перебір

28 показників із кроком $\Delta_t = 0,1$ [10] Кількість членів вибірки встановлено $k=5$. При заданих умовах максимально можливе значення ЗВ $Q=0,88$, тобто існують набори показників, які забезпечують досягнення вказаного рівня.

При виборі варіанту зміни показників ЗВ, який забезпечить найменшу трудомісткість, доцільно розглядати ті набори, в яких найбільші за вагою показники мають менше початкове значення та мінімальну величину зміни.

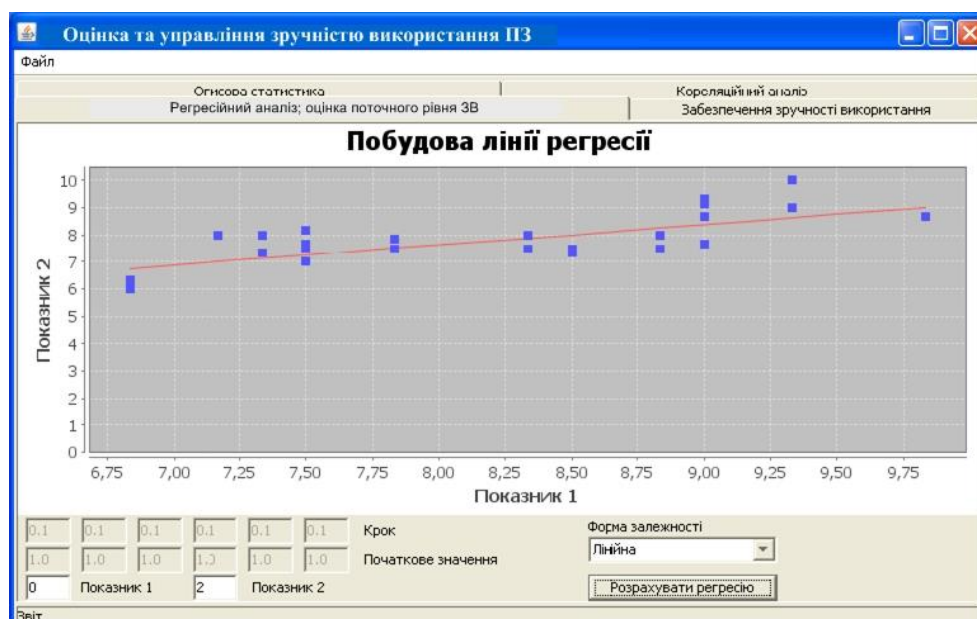


Рис.8. Дослідження форми залежності між показниками ЗВ

Таблиця 1 Емпіричні залежності між показниками ЗВ web-браузеру Chromium 12

Показник 1, (x)	Показник 2, (y)	Коефіцієнт рангової кореляції, r	Формула залежності	Коефіцієнт детермінації, R ²
Часова характеристика	Мінімальна дія	0,71	$y = 6161,2 - 4301,2 \cdot x + 1177,8 \cdot x^2 - 158,5 \cdot x^3 + 10,5 \cdot x^4 - 0,3 \cdot x^5$	0,78
Мінімальне навантаження пам'яті	Оперативність	0,7	$y = 1980,9 - 984,8 \cdot x + 182,5 \cdot x^2 - 14,9 \cdot x^3 + 0,5 \cdot x^4$	0,66
Навігація	Узгодженість	0,8	$y = -47,3 + 20 \cdot x - 2,5 \cdot x^2 + 0,1 \cdot x^3$	0,72
Керування користувачем	Інформативність	0,84	$y = 1963,2 - 1458,3 \cdot x + 420 \cdot x^2 - 58,8 \cdot x^3 + 4 \cdot x^4 - 0,1 \cdot x^5$	0,76
Зручність читання	Повнота	0,79	$y = 7312,7 - 4691,9 \cdot x + 1192,8 \cdot x^2 - 150,2 \cdot x^3 + 9,4 \cdot x^4 - 0,2 \cdot x^5$	0,74

Також бажано, щоб сумарна зміна показників з набору була мінімальна. Тобто перевіряється виконання наступних умов:

$$\begin{cases} A_{i_0} \rightarrow \min, \\ p_{i_{\max}} \rightarrow \max, \\ \sum_{i=1}^5 \Delta_i \rightarrow \min, \end{cases} \quad (1)$$

де A_{i_0} (initial indicator rate) - початкове значення показника (до змін), який має максимальну вагу в даному варіанті;

$p_{i_{\max}}$ (maximal indicator weight) - максимальна вага показника в поточному варіанті;

$\sum_{i=1}^5 \Delta_i$ (sum of the changes) - це сума змін показників в розглядуваному варіанті, яка забезпечує досягнення заданого рівня зручності використання.

У рамках дослідження було встановлено, що більшість варіантів змін мало відрізняються один від одного за вищевказаними параметрами. Максимально задовольняють умові (1) наступні значення: $\sum_{i=1}^5 \Delta_i = 0,95$ для варіанту, зображеному на рис.9а, і $\sum_{i=1}^5 \Delta_i = 1,03$ для рис.9б.

Максимальна вага показника

$$0,21 \leq p_{i_{\max}} \leq 0,23$$

Порівнюючи вказані варіанти, можна зробити висновок, що вибір одного з них доцільно здійснювати за трудомісткістю покращення показника «Відгук» web-браузеру Chromium 12. Зауважимо, що величини усіх показників ЗВ є досить великими, а для отримання вказаного рівня ЗВ мають набувати значень близьких до 1. З останнього впливає значна трудомісткість будь-яких змін.

Всередині більшості наборів можливі значення показників варіюються несуттєво. Найнижчу оцінку мають наступні показники: мінімальне навантаження пам'яті – 0,77; відгук – 0,75; керування користувачем – 0,75; гнучкість – 0,78. Відповідно, у більшості варіантах досягнення заданого рівня ЗВ присутні саме вони.

Слід зазначити, що для дослідження можливості збільшення рівня ЗВ $Q > 0,88$, необхідно встановити кількість членів вибірки $k > 5$.

У рамках експериментальної перевірки роботи програмної системи не ставилась задача покращення показників ЗВ досліджуваного ПЗ та перерахунку оцінки ЗВ на наступній ітерації. Це пояснюється тим, що модель оцінки та управління ЗВ ґрунтується на формулі адитивної згортки, і, очевидно, що зміна показників згідно обраного експертами варіанту покращення ЗВ та підтвердження величини цієї зміни оцінками користувачів при наступній ітерації, забезпечує отримання заданого рівня ЗВ.

Висновки

Для реалізації запропонованого в попередніх дослідженнях методу визначено функціональні вимоги та розроблена архітектура програмної системи, що дозволяє автоматизувати процес оцінки та управління ЗВ при гнучкій розробці програмного забезпечення.

Архітектура поєднує в собі елементи клієнт-серверної архітектури та архітектури, орієнтованої на БД. Система складається з трьох основних частин: Web-сервіс, що забезпечує збір даних; база даних, яка зберігає зібрану інформацію та налаштування системи; аналізатор, що забезпечує управління системою, аналіз даних та виведення результатів.

Для оцінки працездатності розробленої програмної системи і, безпосередньо, запропонованого методу оцінки та управління зручністю використання програмного забезпечення було зібрано та досліджено дані опитування користувачів та експертів на прикладі web-браузеру Chromium 12, а саме:

1. Дана коротка характеристика деяких отриманих метрик ЗВ для web-браузеру Chromium 12.
2. Отримана оцінка поточного рівня ЗВ web-браузеру Chromium 12.
3. Виявлено залежності між показниками ЗВ, а також їх форми. Зроблено висновок щодо можливості впливу зміни одного показника на інші.
4. Досліджено варіанти управління ЗВ web-браузеру Chromium 12 для значення ЗВ $Q = 0,88$ та кількості змінюваних показників $k=5$.

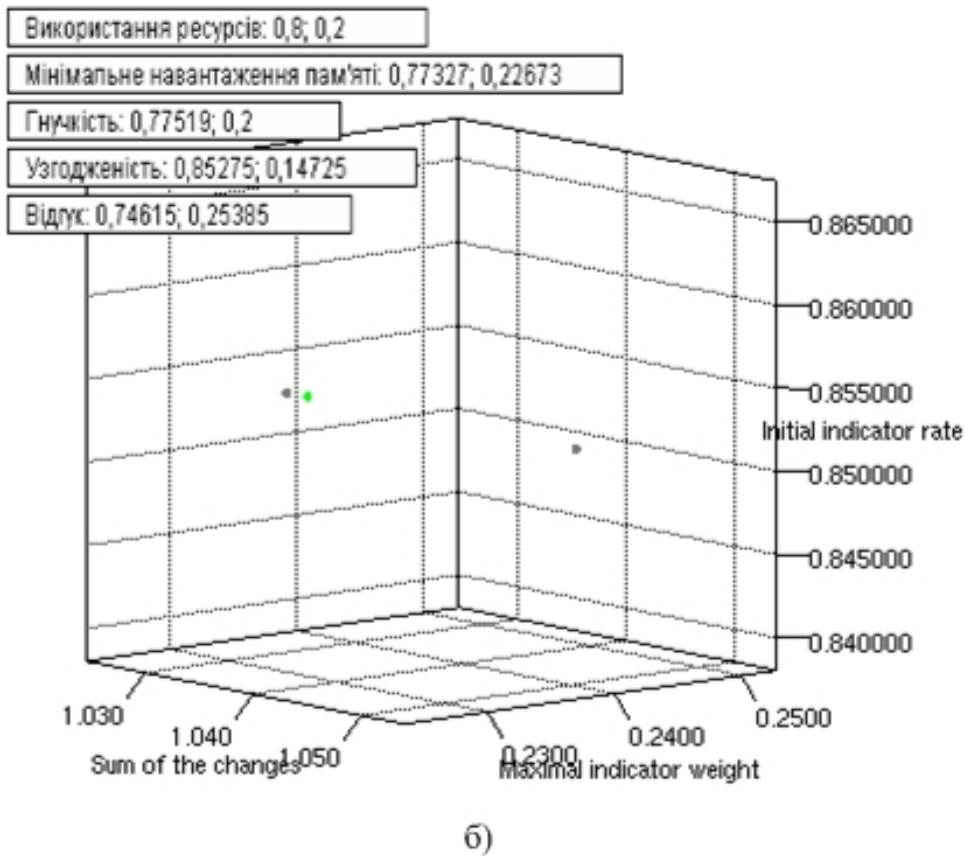
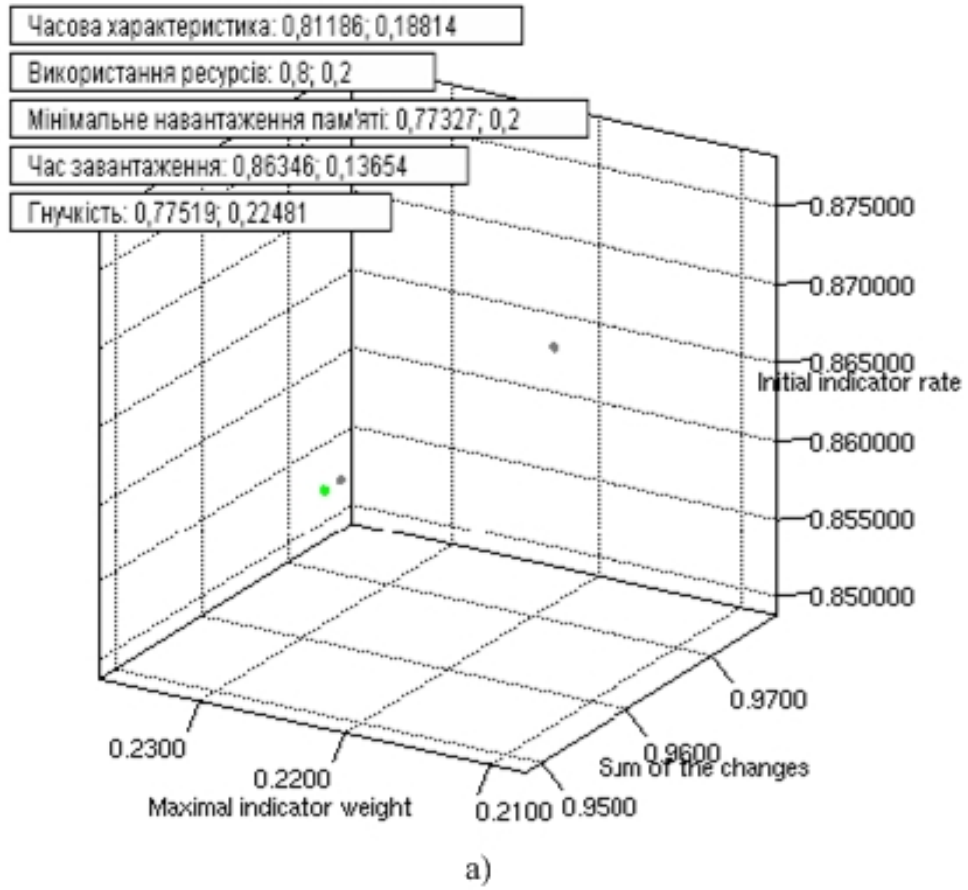


Рис.9. Варіанти зміни показників ЗВ web-браузеру Chromium 12

Список літератури

1. *Folmer E.* Architecting for usability: a survey / E. Folmer, J. Bosch // *The Journal of Systems and Software*. – 2004. – №70 (1-2). – pp. 61-78.
2. IEEE Standard Glossary of Software Engineering Terminology: IEEE Std. 610.12-1990. – Los Alamitos, CA: Institute of Electrical and Electronics Engineers, 1990. – 84 p.
3. Information Technology, Software Product Evaluation, Quality Characteristics and Guidelines for their Use: ISO/IEC 9126. – Geneva: International Organization for Standardization /International Electrotechnical Commission, 1991.
4. Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 11: Guidance on Usability: ISO 9241-11. – Geneva: International Organization for Standardization, 1998.
5. Software Engineering, Product Quality, Part 1: Quality Model: ISO/IEC 9126-1. – Geneva: International Organization for Standardization /International Electrotechnical Commission, 2001. – 25 p.
6. *Padda Harkirat.* QUIM: A Model for Usability/Quality in use Measurement. Lambert Academic Publishing. – 2010. – 124 p.
7. *Сидоров М.О., Гученко І.В.* Оцінка зручності застосування програмного забезпечення в контексті Agile-розробки. Наукоємні технології. – К.: НАУ-друк, 2009. – № 4(4). – С.64-68.
8. *Гученко І.В.* Зручність застосування програмного забезпечення: атрибути та метрики. Інженерія програмного забезпечення. – К.: Бланкодрук, 2010. – №2. – С. 5-13.
9. *Гученко І.В.* Аналіз експертної бази на узгодженість методами Data Mining. Вісник Київського Національного університету ім. Тараса Шевченка. Серія: Фізико-математичні науки. – К.: ДП «Інформаційно-аналітичне агентство», 2011. - №1. – С. 119-122.
10. *Гученко І.В.* Математична модель забезпечення та оцінки практичності програмного забезпечення. Вісник НАУ. – К.: Вид-во Національного авіаційного університету «НАУ-друк», 2011. – №4.
11. IEEE Standard for a Software Quality Metrics Methodology: IEEE Std 1061-1998. – N.Y.: The Institute of Electrical and Electronics Engineers, 1998.
12. *Ivory M., Hearst M.* The State of the Art in Automating Usability Evaluation of User Interfaces // University of California, Berkeley ACM Computing Surveys. – 2001. – Vol. 33, No. 4. – p.470-516
13. *Balbo S.* Automatic evaluation of user interface usability: Dream or reality // Proceedings of the Queensland Computer-Human Interaction Symposium. Bond University. – 1995. – p.478 - 487
14. *Hussain Z., Slany W., Holzinger A.* Current State of Agile User-Centered Design: A Survey // HCI and usability for E-inclusion. Lecture Notes in Computer Science, SpringerLink. – 2009. – Vol. 5889. – p. 416-427.
15. *John B., Kieras D.* Using GOMS for user interface design and evaluation: Which technique? // ACM transactions on Computer-Human interaction. – 1996. – Vol. 3. – p. 287-319.
16. *Kieras D., Meyer D.* An overview of the EPIC architecture for cognition and performance with application to human-computer interaction // Human-Computer Interaction. – 1997. – Vol. 12. – p. 391-438.
17. *Sears A.* AIDE: A step toward metric-based interface development tools // Proceedings of the ACM Symposium on User Interface Software and Technology, New York, ACM Press. – 1995. – p. 101-110.
18. *Macleod M., Rengger R.* The development of DRUM: A software tool for video-assisted usability evaluation // People and Computers VIII. Proc. of HCI'93 Conf., Loughborough UK, Cambridge. – 1993. – p. 293-309.
19. *Вигерс К.* Разработка требований к программному обеспечению/Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2004. – 576с.:ил.
20. *Berson A.* Client/Server Architecture. – New York: McGraw-Hill. – 1996. – 569p.
21. *Babers C.* Architecture Sourcebook Vol.2: Data-Centric Architectures. – 1997. – 302p.
22. <http://tomcat.apache.org/>
23. <http://www.mysql.ru/cgi-bin/download/mysql-essential-5.1.53-win32.msi>
24. *Shull F.* Guide to Advanced Empirical Software Engineering / Shull Forrest, Singer Janice, Sjoberg J.K. Dag. – London: Springer-Verlag, 2008. – 388p.
25. <http://techcrunch.com/2011/08/08/whats-the-fastest-web-browser-in-the-real-world-chrome/>

Відомості про автора:



Гученко Інна Володимирівна, асистент, кафедра інженерії програмного забезпечення, Національний авіаційний університет, наукові інтереси – оцінка зручності використання, якість програмного забезпечення, UML та проектування програмного забезпечення.
E-mail: inna.zapozhets@livenau.net