

ОЦІНКА ЗРУЧНОСТІ ЗАСТОСУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОНТЕКСТІ AGILE-РОЗРОБКИ

Сидоров М. О., Гученко І. В.

Національний авіаційний університет

nikolay.sidorov@livenau.net

У статті наведено метод автоматизованої оцінки зручності використання програмної системи при agile-розробці з метою врахування та прогнозування вимог користувачів. Ця методика дає змогу не лише оцінювати поточний рівень зручності застосування програмних систем, але й керувати якістю процесів їхнього створення на основі математичних оптимізаційних моделей.

The article is dedicated to designing of a technique for usability automated evaluation under agile software design for the purpose of taking into account and predicting user's requests. Specified technique allows not only to estimate a current usability level of software systems, but also to manage quality of their creation processes on the basis of mathematical optimization models.

Вступ

В умовах постійно зростаючої конкуренції між розробниками все більше виробників програмного забезпечення вважають зручність використання (*usability*) їхніх продуктів стратегічною ціллю. Однак на практиці в процесі розробки некоординовані дії в поліпшенні зручності мають малу ефективність [1].

Зважаючи на поширення *agile*-методів в інженерії програмного забезпечення, багато експертів з інженерії зручності та проектування інтерфейсу користувача розвинули так звану *agile* людино-орієнтовану інженерію програмного забезпечення (*agile human-centered software engineering*) [2]. Вона наголошує на орієнтованому на користування (*usage-centered design*) підході до розробки програмного забезпечення, підкреслюючи недоліки надмірного фокусування на досвіді та побажаннях користувача (*user-centered design*). Але в обох випадках споживачі залишаються джерелом інформації та валідації. Оцінка зручності використання — важлива частина процесу розробки інтерфейсу користувача, який складається з повторних циклів проектування, моделювання (прототипування) та оцінки [3].

Виходячи з вищенаведеного, можна зробити висновок, що для ефективного управління проектами в області *agile*-розробки програмного забезпечення необхідно перебувати в тісному контакті з користувачами, збираючи їх відгуки щодо зручності використання і, по можливості, прогнозуючи вимоги [4].

Agile-методам і досі бракує інтеграції зручності використання і *agile*-процесів. Розробка невеликими кроками дає коректні результати і майже не фокусується на проблемі зручності використання. Адже зміни в архітектурі програмного забезпечення зазвичай не впливають на те, що бачить та з чим взаємодіє користувач, із зручністю використання — навпаки.

Тому в *agile*-розробці бракує орієнтації на користувача, що суттєво впливає на якість кінцевого продукту [2].

Отже, важлива обізнаність інженерів-практиків програмного забезпечення щодо різних методів оцінки зручності використання та застосування їх відповідно до особливостей проекту [5]. Існуючі дослідження з інтеграції ітераційних методів та орієнтованого на користувача підходу до розробки програмних засобів виокремлюють близько 14 основних методів, найбільш поширені з яких — низькорівневе прототипування, концептуальний дизайн, спостереження за користувачами, оцінка зручності застосування експертами, дослідження в реальних умовах, швидке ітеративне тестування, анкетування, перевірка зручності використання в лабораторних умовах, аналіз потреб, автоматизована оцінка зручності застосування [6; 7]. Однак малодослідженими залишаються методи і програмні рішення, орієнтовані не лише на оцінку поточного рівня зручності використання програмної системи, але й на керування нею на основі відгуків користувачів.

У зв'язку з цим актуальним є залучення нових методів розробки і оцінки зручності використання як важливої частини проектування інтерфейсу користувача у відповідності до вимог стандарту ISO 9001:2000.

Ітераційний метод оцінки зручності застосування, орієнтований на формальний аналіз відгуків користувачів

Запропоновано метод оцінки зручності застосування (*usability*) програмного забезпечення, який передбачає оцінку поточного рівня зручності використання на кожній ітерації в процесі *agile*-розробки та формування оптимальної стратегії досягнення бажаного рівня (який задається розробником на початку).

Для оцінювання зручності застосування слід виконати такі завдання :

1. Побудова експертами ієрархічної моделі показників зручності, про значення яких може судити користувач.

2. Побудова єдиної оцінки зручності застосування програмного забезпечення. Кожному показнику в рамках ієрархічної моделі приписується числова величина — «значимість показника». Це дає змогу відповідно до ієрархічної моделі, доповненої операторами перетворення, зводити окремі значення показників зручності в єдину характеристику.

3. Побудова математичної моделі забезпечення зручності застосування. Побудована математична модель доповнюється функцією оцінки трудомісткості зміни показників зручності, тим самим, отримуючи модель забезпечення необхідного її рівня та застосування оптимальним чином.

4. Отже, завдання зводиться до оптимізаційної задачі при мінімізації трудомісткості робіт для забезпечення заданого рівня зручності.

5. Формування показників зручності на підставі відгуків. Для формування поточних значень показників зручності застосування необхідно проведення опитування користувачів, які дають числову оцінку за кожним показником.

6. Формування оптимальної стратегії. У разі узгодженості оцінок користувачів можлива побудова оптимальної стратегії поліпшення за рахунок підставлення значень показників зручності в модель забезпечення необхідного рівня зручності застосування та відшукування оптимального розв'язання. Вирішенням є набір показників, що потребують поліпшення (з урахуванням величини зміни кожного показника). Для того, щоб визначити ключові напрями поліпшення, кількість показників, що потребують його, штучно обмежується.

7. Впровадження обраної стратегії, контроль виконання та коригування моделі.

На рис. 1 подано діаграму діяльності, що ілюструє цей метод.

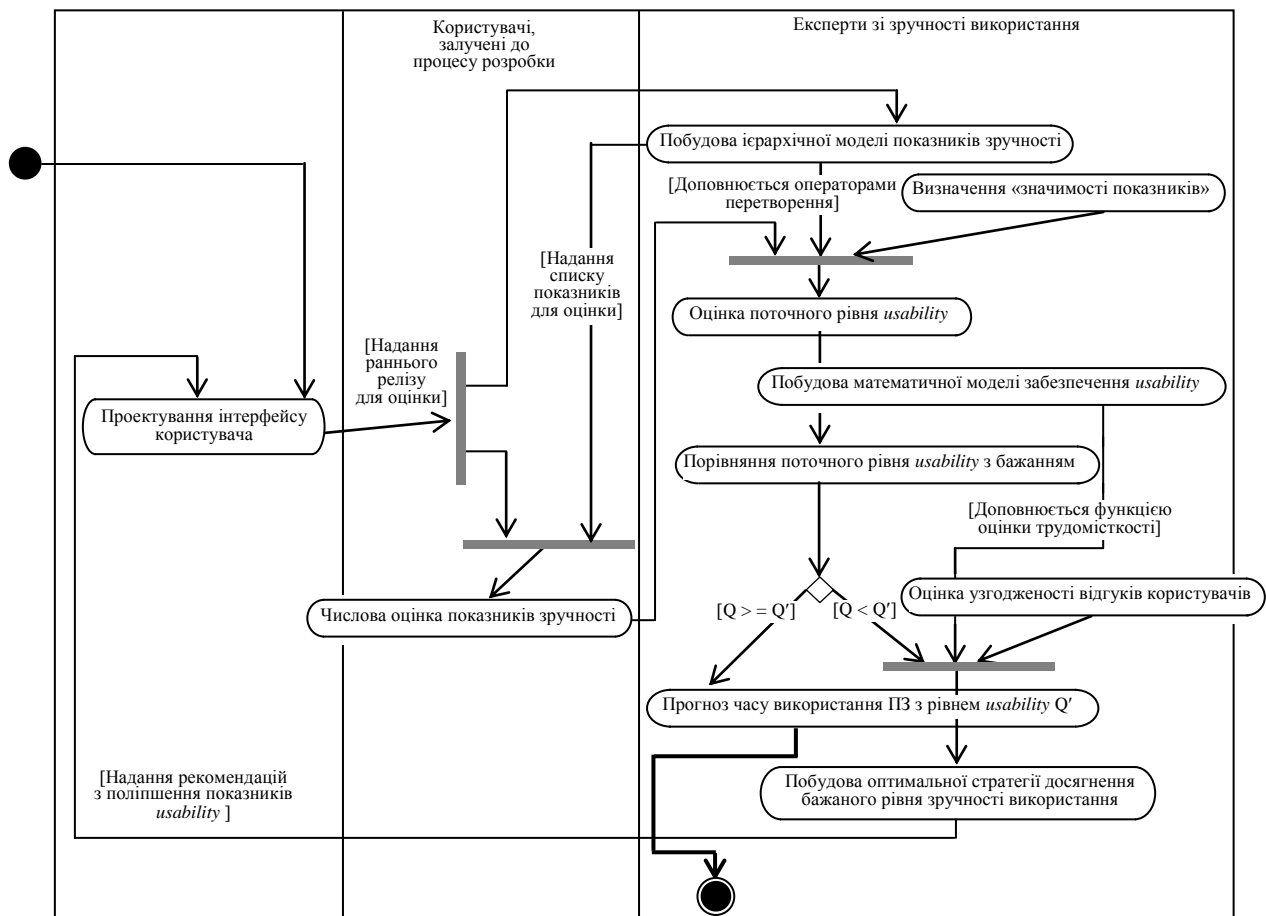


Рис. 1. Діаграма діяльності «Ітераційна оцінка зручності застосування програмної системи в процесі agile-розробки»

Для підтримки програмними засобами вищенаведений метод зручно представити у вигляді п'яти модулів:

1) модуль збору даних, формування значень показників зручності та єдиної оцінки зручності використання;

2) модуль побудови зв'язків усередині математичної моделі забезпечення та контролю зручності застосування програмної системи;

3) модуль побудови прогнозів змін показників зручності;

4) модуль аналізу експертної інформації оцінки узгодженості експертних оцінок;

5) головний модуль оцінки зручності застосування та побудови оптимальної стратегії забезпечення необхідного рівня зручності.

I модуль. Для забезпечення та оцінювання зручності застосування необхідно зібрати та проаналізувати дані щодо ступеня «вдоволеності» користувачів. За основу цілком логічно взяти проведення масового опитування. Доцільно зосередити увагу на формуванні значень показників. Отже, виникає питання, якими формулами користуватись для обчислення середніх величин? Зазвичай використовують середнє арифметичне. Але давно вже відомо, що такий спосіб неко-ректний, оскільки бали зазвичай вимірюються за порядковою шкалою. Обґрунтованим є використання медіан. Однак виправданим є одночасне застосування обох методів — методу середніх арифметичних рангів (балів), і методу медіанних рангів. Така рекомендація відповідає концепції стійкості, яка пропонує використовувати різні методи для обробки одних і тих самих даних з метою виявити висновки, що є спільними для всіх методів (що зменшує факт суб'єктивності дослідника) [8].

Для представлення моделі оцінки зручності використаємо нотацію семантичної мережі [9].

Семантична мережа відображає сукупність об'єктів предметної області і відношень між ними, при цьому об'єктам відповідають вершини (або вузли) мережі, а відношенням — дуги, що сполучають їх. Як об'єкти (вершини) виступатимуть показники різних рівнів тлумачення поняття зручності застосування (рис. 2).

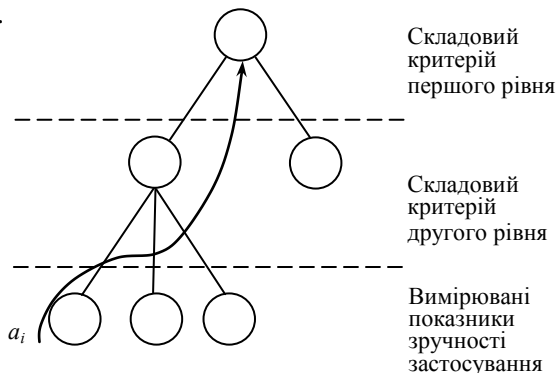


Рис. 2. Графічне представлення ієрархії складових критеріїв

Значення a_i формуються на основі відгуків користувачів програмного забезпечення.

II модуль. Після певного ланцюгу формалізації, як модель забезпечення зручності застосування програмної системи розглядатимемо n -ви-мірний простір (n — кількість показників) від поточного значення до 1 (абсолютного значення) з кроком Δ . У вузлах задано: рівень зручності у певний момент часу — q , а також значення трудомісткості забезпечення заданого рівня у вузлі — h .

Задача забезпечення необхідного рівня зручності застосування Q' зводиться до відшукування вершини, для якої:

$$q \geq Q', \forall (q', h'): h' < h \Rightarrow q' < Q'.$$

Розв'язком буде набір параметрів, що являють собою значення поліпшення по кожному показнику.

Наступне важливе питання — методика зниження розмірності задачі. Воно пов'язане з побудовою явного виду залежностей між змінними. Дуже часто використовують метод найменших квадратів, але він є досить грубим наближенням. Алгоритми Лагранжа та Невілля мають свої недоліки в практичній реалізації програмними засобами та швидкодії.

З нових досягнень в області поліноміальної інтерполяції зазначимо методи, які ґрунтуються на так званому барицентричному поданні інтерполяційного полінома:

$$r(x) = \frac{\sum_{j=0}^N \frac{w_j}{x - x_j} y_j}{\frac{w_j}{x - x_j}}, \quad w_j = \frac{1}{\prod_{k \neq j} (x_j - x_k)}.$$

У загальному випадку ця формула задає раціональну функцію, що проходить через точки (x_j, y_j) , незалежно від того, які вагові коефіцієнти w_j ми обираємо. Таким чином, ці коефіцієнти впливають лише на те, як інтерполянт наближає функцію між точками [10].

III модуль. Алгоритм побудови прогнозів є модифікацією алгоритму знаходження залежностей між змінними лише з тією відмінністю, що однією з досліджуваних змінних є час. Це дасть змогу визначати функцію наближення у будь-якій точці часового інтервалу (як «у минулому», так і «в майбутньому»).

IV модуль. Під час оцінювання зручності застосування для отримання формальної постановки задачі задіяні експертні оцінки для обчислення ваги та значень показників. При цьому важливою є перевірка експертних оцінок на узгодженість. Для цього пропонується застосувати метод узгодження кластеризованих ранжувань. Цей метод кращий за обчислення коефіцієнтів конкордації Кендела (що досить часто використовується), оскільки дозволяє

визначити те загальне, що знаходиться у первинних ранжуваннях.

Нехай маємо скінченне число об'єктів (для простоти зображатимемо натуральними числами), що називають «носієм». Кластеризованим ранжуванням, визначеним на даному носії, називають наступну математичну конструкцію [11]:

Нехай об'єкти розбиті на групи — кластери. У кластері може бути від одного до n об'єктів (n — натуральне).

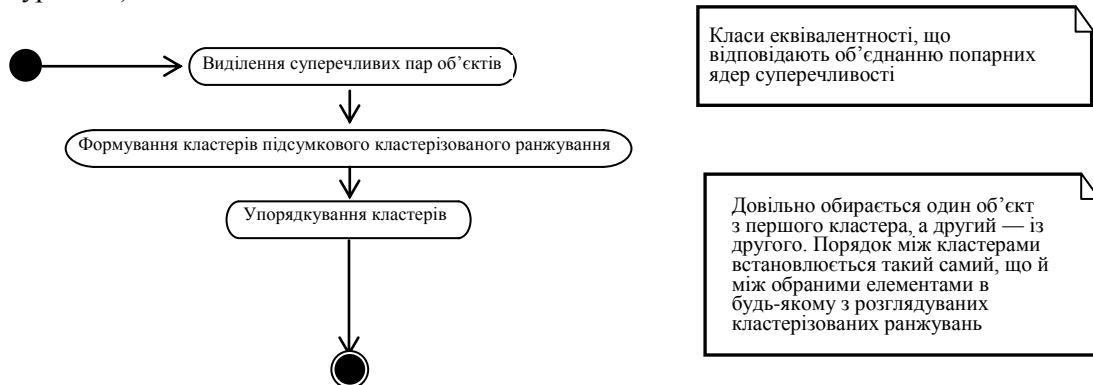


Рис. 3 Алгоритм узгодження кластеризованих ранжувань

Наведемо приклад

Нехай потрібно перевірити на узгодженість думки експертів щодо важливості атрибутів такої характеристики якості програмного продукту, як зручності.

Позначимо розглядувані атрибути через A, B, C, D і E , а думки трьох експертів — L, M та N відповідно.

1. Кластеризовані ранжування з суворим лінійним порядком мають вигляд:

$$L = [A < C < B < E < D];$$

$$M = [A < C < E < B < D];$$

$$N = [A < E < B < C < D].$$

2. Ядра суперечності для кожної пари ранжувань мають вигляд:

$$S(L, M) = [(B, E)];$$

$$S(L, N) = [(B, C), (C, E), (B, E)];$$

$$S(M, N) = [(B, C), (C, E)].$$

3. Результат узгодження ранжувань такий:

$$f(L, M) = [A < C < \{B, E\} < D];$$

$$f(L, N) = f(M, N) = f(L, M, N) =$$

$$= [A < \{B, C, E\} < D].$$

Тобто всі експерти вважають атрибут A за найважливіший, а атрибут D — найменш гідним уваги. До того ж видно, що думка експерта N відрізняється найсуттєвіше, що вимагає з'ясування питання про його кваліфікацію та причини таких відмінностей.

У модуль потребує виконання таких дій:

1. Оцінка поточного показника зручності на основі введених показників (отриманих з відгуків користувачів): кожне зі значень показників зручності застосування перемножується на вагу показника (що

Наприклад, об'єкти $1, 2, 3, \dots, 10$ можуть бути розбиті на 7 кластерів: $\{1\}$, $\{2, 3\}$, $\{4\}$, $\{5, 6, 7\}$, $\{8\}$, $\{9\}$, $\{10\}$. Кластери не мають спільних елементів, а об'єднання їх (як множин) і становить усю множину. Між кластерами встановлено суворий лінійний порядок.

Алгоритм узгодження деякого числа кластеризованих ранжувань [11] подано на рис. 3.

визначається за допомогою експертів) і, таким чином, отримуємо підсумкову оцінку зручності.

2. Вибір стратегії забезпечення необхідного рівня зручності застосування.

3. Отримання результатів (значень необхідного поліпшення за кожним показником).

Висновки

Agile методи розробки програмного забезпечення є гнучкими, ітеративними та «легкими», тому досить зручно інтегрувати в них методи забезпечення зручності використання. Це досягається за рахунок спрямованості на користувачів та безупинного тестування. Результати опитувань показують, що більшість інженерів—розробників зазначають позитивний вплив такої інтеграції на поліпшення якості розроблюваного програмного продукту, зростання задоволення кінцевих користувачів.

Запропонований метод оцінювання зручності застосування програмного забезпечення дає змогу не лише оцінювати поточний рівень зручності під час кожної ітерації, але й керувати зручністю використання створюваних програмних продуктів на основі математичних оптимізаційних моделей.

Запропонований метод дає можливість достовірної оцінки бізнес-логіки та інтерфейсної прийнятності (зручності), що істотно впливає на загальний рівень якості програмного забезпечення.

Таким чином, описаний метод є засобом інтеграції розробки, орієнтованої на користувача, та *agile*-методів.

Подальше дослідження у цьому напрямку доцільно спрямувати на аналіз створених математичних моделей забезпечення та оцінки зручності програмного забезпечення на предмет адекватності, можливості програмної реалізації та оптимізації обчислень.

Також потребують дослідження та деталізації методики збору та аналізу даних, стратегії забезпечення необхідного рівня зручності застосування тощо.

ЛІТЕРАТУРА

1. *Granollers T., Lores J. & Perdrix F.* Usability Engineering Process Model. Integration with Software Engineering, Proceedings of HCI International 2003, Crete, Greece, June 22-27-2003, Lawrence Erlbaum Associates, New Jersey, USA.
2. *Memmel T., Gundelsweiler F., Reiterer H.* Agile Human-Centered Software Engineering. British Computer Society Conference on Human-Computer Interaction archive. Proceedings of the 21st British CHI Group Annual Conference on HCI 2007, University of Lancaster, United Kingdom, Vol. 1. — Pp. 167—175.
3. *Dyba T., Dingsoyr T.* Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50(9-10) (August 2008). — Pp. 833—859.
4. *Holzinger A.* Usability Engineering for Software Developers. *Communications of the ACM* 48(1) (January 2005). — Pp. 71—74.
5. *Melody Y. Ivory and Marti A. Hearst.* The State of the Art in Automating Usability Evaluation of User Interfaces// *ACM Computing Surveys*. — Vol. 33, No. 4, December 2001. — Pp. 470—516.
6. *Vredenburg K., Mao J.Y., Smith P.W., Carey T.*: A survey of user-centered design practice. In: CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, 2002.
7. *Zhou R., Huang S., Qin X., Huang J.*: A survey of user-centered design practice in China. In: IEEE International Conference on Systems, Man and Cybernetics, SMC, 2008. — Pp. 1885—1889.
8. *Орлов А. И.* Нечисловая статистика / А. И. Орлов. — М. : МЗ-Пресс, 2004. — 513 с.
9. *Саати Т.* Принятие решений. Метод анализа иерархий / Т. Саати. — М. : Радио и связь, 1993.
10. *Jean-Paul Berrut, Lloyd N. Trefethen.* Barycentric Lagrange Interpolation/ *SIAM REVIEW*/ Vol. 43, 2004, No. 3. — Pp. 501— 517
11. *Горский В. Г.* Метод согласования кластеризованных ранжировок // *Автоматика и телемеханика* / В. Г. Горский, А. И. Орлов, А. А. Гриценко. — 2000. — № 3. — С. 159—167.

Стаття надійшла до редакції 16.01.10.