УДК 004.41

# ЗАСТОСУВАННЯ КЛАСТЕРИЗАЦІЇ ДЛЯ ПЛАНУВАННЯ РЕЛІЗІВ

**В. В. Любченко**
*Одеський національний політехнічний університет*

***Анотація.*** *Розглянуті задача планування релізів та її окремий випадок – задача наступного релізу. Запропоновані міри схожості для рішення задачі планування релізів за допомогою методів кластеризації. Проаналізовані результати планування релізів на основі кластеризації вимог. Зроблено висновок про необхідність додаткових мір схожості.*

***Ключові слова:*** *планування релізів, задача наступного релізу, кластеризація, міра схожості.*

## Вступ

Однією з ключових проблем проектів з розробки програмних продуктів є проблема забезпечення якості результатів розробки. Відповідно до визначення ISO якість – це повнота властивостей і характеристик продукту, процесу або послуги, яка забезпечує здатність задовольняти заявленим або очікуваним вимогам.

Відповідно до стандарту ISO 25010 загальне подання якості програмного продукту рекомендовано описувати трьома взаємодіючими і взаємозалежними моделями якості, що відображують

- якість продукту, задану вимогами замовника в специфікаціях і віддзеркалену в характеристиках кінцевого продукту;
- якість даних;
- якість при використанні в процесі нормальної експлуатації і результативність досягнення потреб користувачів з урахуванням витрат ресурсів.

Остання модель відповідає найбільш впливовим факторам, що визначають успішність програмного продукту [1]. Через специфічну природу програмного продукту якість при використанні можна оцінити лише після завершення розробки. Цей чинник обумовив широку розповсюдженість ітеративної розробки програмного забезпечення (ПЗ).

Ітеративна розробка – це методологія, в якій послідовно повторюються процеси з розробки програмного продукту і безперервного аналізу отриманих результатів з коригуванням наступних етапів роботи. На початку розробки програмного продукту складається специфікація вимог до нього, до якої на протязі розробки можуть вноситися зміни та доповнення. Задача особи, яка приймає рішення (ОПР), полягає в визначенні на початку кожної ітерації підмножини вимог, які мають розроблятися на наступній ітерації для того, щоб задовольнити найбільше потреб споживачів за умови мінімізації витрачених коштів розробника ПЗ [2]. Результатом виконаних на протязі чергової ітерації робіт є наступний реліз продукту. Визначення змісту (scope of work – SOW) кожного релізу в плановану періоді є по суті кластерізацією вимог до продукту (features) в часі.



Планування релізів є важливим етапом інженерії вимог до продукту. Воно є складною задачею через необхідність розуміння і врахування цілей планування і технічних обмежень, які необхідні для хорошого плану релізів [3]. Незважаючи на очевидну необхідність планування релізів, методології, які застосовуються для цього зараз, базовані в основному на якісних інструментах і не забезпечують кількісні засоби для планування.

Метою роботи є дослідження властивостей застосування кластеризації як засобу планування релізів на основі кількісних характеристик вимог до ПЗ.

## 1. Задача планування наступного релізу

Визначення вимог для майбутніх релізів є складним процесом. Типовим випадком є прагнення реалізувати як можна більше властивостей в ранніх релізах. Проте бажану кількість вимог обмежує кількість наявних ресурсів. Крім того, вимоги до ПЗ не є незалежними, і в процесі планування слід враховувати технологічні залежності. Отже існує необхідність враховувати цінність включення певної вимоги до наступного релізу порівняно з цінністю інших вимог, що потребують той самий ресурс. Задачу планування, яка полягає в тому, щоб вирішити, які вимоги включити до наступного релізу, називають задачею наступного релізу (Next Release Problem – NRP).

Оскільки NRP є різновидом аналізу зисків і витрат, її рішення має бути оптимальним за Парето [4]. В цьому випадку NRP формулюють як задачу пошукової оптимізації [5].

Хай є множина споживачів ПЗ $C = \{c_1, ..., c_m\}$, кожен з яких характеризується важливістю для компанії, що віддзеркалюється множиною вагових коефіцієнтів $Weight = \{w_1, ..., w_m\}$, де $w \in [0,1]$ та $\sum_{j=1}^{m} w_j = 1$. Припустимо також, що є множина вимог $R = \{r_1, ..., r_n\}$, які треба задовольнити в наступних релізах існуючого ПЗ. Програмна реалізація кожної вимоги потребує певних ресурсів, які можуть бути оцінені в термінах вартості $Cost = \{cost_1, ..., cost_n\}$.

Реалізація вимог забезпечує зиски для компанії. Рівень задоволення окремих споживачів залежить від того, яка підмножина вимог буде реалізована в наступному релізі програмного продукту. Важливість вимог не однакова для різних споживачів. Кожен споживач $c_j$ ($1<j<m$) встановлює для вимоги $r_i$ ($1<i<n$) значення $value(r_i, c_j)$, при цьому $value(r_i, c_j)>0$, коли споживач $j$ зацікавлений у вимозі $i$, та 0 – в іншому випадку.

Вектор рішення $\vec{x} = (x_1,...,x_n)$, в якому $x_i \in \{0,1\}, i = \overline{1,n}$, визначає, які з вимог будуть обрані в розробку на наступній ітерації. Елемент $x_i$ дорівнює 1, якщо вимога $i$ включена до розробки на наступній ітерації, та дорівнює 0 в іншому випадку.

Очевидно, що при рішенні NRP слід взяти до уваги дві цільові функції. Перша з них відповідає загальному зиску для компанії, якій очікується отримати як результат $k$-ї ітерації:

$$f_1(\vec{x}) = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} w_j \cdot value^k(r_i, c_j).$$

Задача полягає у виборі підмножини вимог, яка призведе до максимізації зиску для компанії.

Друга цільова функція відповідає загальним коштам, потрібним для задоволення вимог на $k$-й ітерації:

$$f_2(\vec{x}) = \sum_{i=1}^{n} cost_i^k \cdot x_i .$$

Задача полягає у виборі підмножини вимог, яка призведе до мінімізації коштів компанії.

Відомо [4], що Парето-фронт визначається як $P(Y) = \{y \in Y : \{y' \in Y : y' \succ y\} = \varnothing\}$, де відношення домінування по Парето $y' \succ y$ означає, що $y \le y', y \ne y'$. Тому задача мінімізації по другому критерію має бути перетворена на задачу максимізації, помножимо цільову функцію на -1. Тепер ЗНР для $k$-ї ітерації можна записати в такому виді:

$$\text{Maximizef}_1(\vec{x}) = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} w_j \cdot value^k(r_i, c_j)$$

$$\text{Maximizef}_2(\vec{x}) = -\sum_{i=1}^{n} cost_i^k \cdot x_i$$

Таким чином NRP є задачею оптимізації з двома цільовими функціями на комбінаторній множині комбінацій вимог, тобто NP-трудною задачею. Останнім часом багато уваги приділяється дослідженням застосування евристичних пошукових технік для її рішення.

Проте рішення класичної NRP має суттєвий недолік, пов'язаний з тим, що формулювання задачі ігнорує залежності між вимогами до ПЗ. З метою подолання цього обмеження у розгляд вводять два типи відношень, які моделюють залежності між вимогами [6]:

– відношення передування $Prec(r_i,r_j)$, яке означає, що вимога $r_i$ має бути реалізована не пізніше вимоги $r_j$;

– відношення одночасності $Coup(r_i,r_j)$, яке означає, що вимога $r_i$ має бути реалізована на тій самій ітерації, що і вимога $r_j$.

Врахування існування залежностей між вимогами призводить до необхідності внесення певних змін в оцінки вартості та зиску, що характеризують вимоги. Проте навіть внесення таких змін залишає модель «реактивною», оскільки вона враховує часові і логічні залежності, які мають місце на момент реалізації вимоги $r_i$, і не враховують факт того, що реалізація вимоги $r_i$ також відкриває можливість реалізації інших вимог.

В галузі розробки ПЗ часто застосовують методи кластеризації для вирішення задач рефлексивного аналізу структури ПЗ, еволюції успадкованого ПЗ та відновлення інформації. NP-трудна задача призначення вимог до релізів також може бути досить просто вирішена за допомогою методів кластеризації. Складність задачі значно зменшується шляхом об'єднання вимог до кластерів «обсягу релізу».

## 2. Задача планування релізів як задача кластеризації

Традиційно рішення задачі кластеризації складається з чотирьох етапів: визначення атрибутів для опису об'єктів кластеризації, вибір алгоритму кластеризації, перевірка достовірності результатів та інтерпретація результатів. Рішення задачі кластеризації вимог до ПЗ дозволить вирішити, які вимоги мають бути реалізовані в якому релізі.

Для визначення схожості вимог в процесі кластеризації рекомендовано застосовувати чо-

тири міри, які враховують існуючі між вимогами залежності [7]. Проте, на відміну від [7], ми пропонуємо визначати їх кількісні вимірювачі на основі операцій над множинами. Розглянемо докладніше ці міри схожості.

Схожість базована на передуванні – найбільш очевидна міра схожості, яка заснована на обмеженнях технологічного впорядкування. Вимога не може бути додана до кластеру (який представляє реліз) до того, як заплановані її попередники.

Несхожість базована на ресурсах – при створенні кластерів важливо врахувати, що вимоги конкурують одна з одною на одному пулі ресурсів. Отже, міра схожості може бути отримана через потребу в ресурсах – чим більше дві вимоги потребують однакових ресурсів, тим менше вони можуть бути кластеризовані разом. Кількісно міра визначається як

$$\rho_{i,j} = 1 - \frac{\left| r_i \cap r_j \right|}{\left| R \right|},$$

де $r_i$ – ресурси, які потрібні для реалізації вимоги $i$, $R$ – доступні ресурси.

Схожість базована на попередниках – інший аспект, що віддзеркалює залежності між вимогами, а саме наявність однакових попередників:

$$\pi_{i,j} = \frac{\left| pred_{i,j} \right|}{\left| pred_i \cup pred_j \cup pred_{i,j} \right|},$$

де $pred_j$ – це множина попередників для вимоги $j$, $pred_{i,j}$ – множина спільних попередників для вимог $i$ та $j$.

Схожість базована на наступниках – останній аспект, що віддзеркалює залежності між вимогами, а саме наявність однакових наступників:

$$\eta_{i,j} = \frac{\left| succ_{i,j} \right|}{\left| succ_i \cup succ_j \cup succ_{i,j} \right|},$$

де $succ_j$ – це множина наступників для вимоги $j$, $succ_{i,j}$ – множина спільних наступників для вимог $i$ та $j$.

В процесі кластеризації запропоновані міри схожості можна використовувати як окремо, так і разом, визначивши на їх основі інтегральний коефіцієнт схожості.

У разі кластеризації вимог складність задачі планування релізів скорочується з $O(n!)$, де $n$ – кількість вимог, до $O(m^2)$, де $m$ – планована кількість релізів.

Звичайно, як результат визначаються кластери, але їх реалізація не розглядається. Проте, для конкретної мети планування релізів існує очевидний критерій для визначення розміру кластера – ресурсні обмеження для конкретного релізу.

Розглянемо на прикладі, як виконується кластеризація вимог та проаналізуємо їх змістовну інтерпретацію.

### 3. Приклад кластеризації вимог для планування релізів

Для вивчення результатів кластеризації для планування релізів використаємо мережу вимог проекту Virtual Innovation Space (рис. 1), який вже було реалізовано, тобто забезпечимо можливість виконати порівняння обчисленого результату з реальністю.

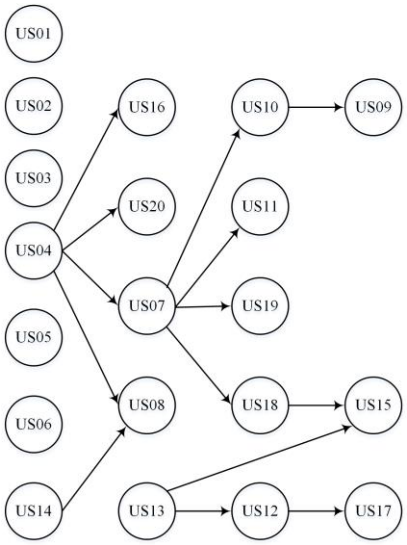


Рис. 1. Мережа вимог проекту

Розглянемо кластеризацію вимог на основі інтегрального коефіцієнта схожості, який враховує схожість базовану на попередниках і схожість базовану на наступниках. Схожість базована на передуванні є окремим випадком міри схожості, яка відіграє роль тригера і є залежною від результатів кластеризації за іншими мірами, тому виключимо її з розрахунку. Інформацію про ресурси залишимо для визначення розміру кластера. Для визначення кластерів застосуємо ієрархічну кластеризацію, результат якої показано на рис. 2. Зауважимо, що кластеризація була виконана лише для вимог, які мають попередників та/або наступників.

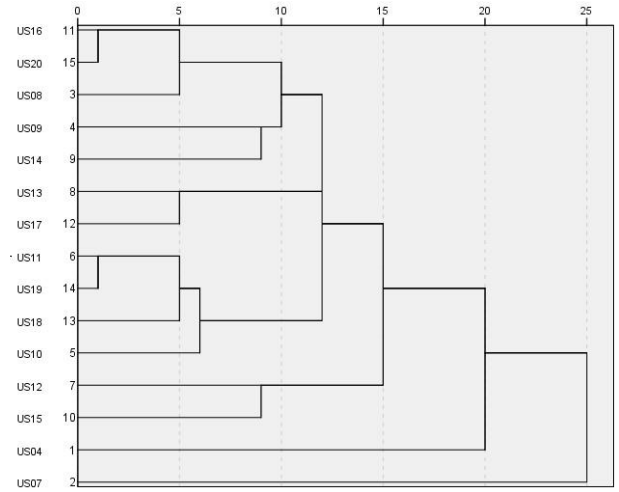


Рис. 2. Дендрограма результатів кластеризації

Припустимо (для спрощення), що всі вимоги потребують однакової кількості ресурсів, а доступні ресурси дозволяють реалізувати по п'ять вимог на протязі однієї ітерації. Розподіл вимог по релізах показано на рис. 3.

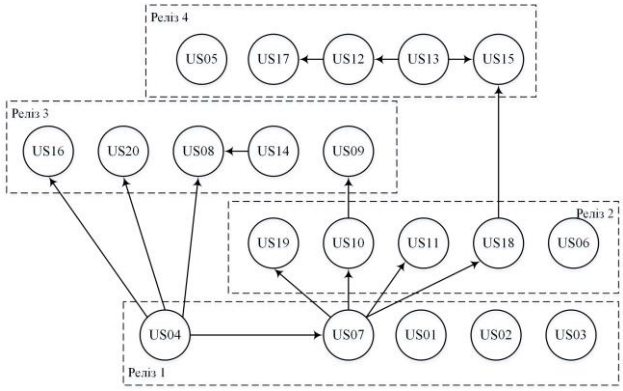


Рис. 3. Розподіл вимог по релізах

Як бачимо, результат кластеризації з використанням мір схожості на основі мережі впорядкування вимог є принципово реалізованим. Певні труднощі можуть викликати ланцюжки залежностей, які заплановані в один реліз (наприклад, US13, US12, US17 на рис. 3). Ця ситуація потребує додаткового вивчення.

Порівняння результатів планування на основі кластеризації з процесом розробки, який було реалізовано, показує, що головним недоліком отриманого результату є невідповідність рекомендованої послідовності до важливості вимог. Таким чином, можемо зробити висновок про доречність розширення характеристик для обчислення схожості вимог таким чином, щоб врахувати їх впорядкування за важливістю або величиною зиску для компанії від їх реалізації.

**Висновки**

В умовах ітеративної розробки ПЗ однією з важливих задач є планування релізів, оскільки після кожного релізу розробники отримують данні про якість свого продукту. Задача планування є NP-трудною задачею, тому для її вирішення застосовують евристичні методи. Проте задача планування може бути вирішена як задача кластеризації, що дозволяє знизити обчислювальну складність рішення.

В роботі розглянуті міри схожості, які можуть бути застосовані при кластеризації, та які віддзеркалюють існуючи між вимогами залежності. Отриманий як результат кластеризації розподіл вимог по релізам є реалізованим, проте залишаються відкриті питання, що потребують додаткового вивчення.

Незважаючи на це можна стверджувати, що методи кластеризації доцільно застосовувати з метою підтримки вирішення задачі планування релізів та NRP.

**Список використаної літератури**

1. Денисюк, А. В. Актуальные проблемы качества программного обеспечения [Текст] / А. В. Денисюк, В. В. Любченко // Электротехнические и компьютерные системы. – 2013. – № 9 (85). – С. 142–148.

2. Durillo, J. J. A study of the bi-objective next release problem [Текст] / J. J. Durillo, Y. Zhang, E. Alba, M. Harman, A. J. Nebro // Empirical Software Engineering. – 2011. – Vol. 16 (1). – P. 29–60.

3. Ruhe, G. The art and science of software release planning [Текст] / G. Ruhe and M. O. Saliu // IEEE Software. – 2005. – Vol. 6, no. 22. – P. 47–53.

4. Ногин, В. Д. Принятие решений в многокритериальной среде: количественный подход [Текст] / В. Д. Ногин. – М. : ФИЗМАТЛИТ, 2002. – 176 с.

5. Harman, M. Search based software engineering [Текст] / M. Harman, B. F. Jones // Information and Software Technology. – 2001. – № 43 (14). – P. 833–839.

6. Любченко, В. В. Система підтримки прийняття рішень для задачі наступного релізу [Текст] / В. В. Любченко // Штучний інтелект. – 2015. – № 1-2 (67-68). – С. 106–110.

7. Etgar, R. Project Scope Partitioning by Clustering Features into Releases of Long R&D Projects [Текст] / R. Etgar, R. Gelbard, Y. Cohen // Procedia Computer Science. – 2016. – Vol. 100. – P. 1235–1241.

**References**

1. Denisiuk, A. V. and Liubchenko, V. V. (2013). Actual problems of software quality [Aktualnye problem kachestva programmnogo

obespecheniia]. *Elektrotekhnicheslie i komputernye sistemy*, 9(85), pp. 142–148.

2. Durillo, J. J., Zhang, Y., Alba, E., Harman, M. and Nebro, A. J. (2011). A study of the bi-objective next release problem. *Empirical Software Engineering*, 16(1), pp. 29–60.

3. Ruhe, G. and Saliu, M. O. (2005). The art and science of software release planning. *IEEE Software*, 6(22), pp. 47–53.

4. Nogin, V. D. (2002). *Decision making in the multicriteria environment: a qualitative approach* [Priniatie reshenii v mnogokriterialnoi srede: kolichestvennyi podkhod]. Moskow: FIZMATLIT, 176 p.

5. Harman, M. and Jones, B. F. (2001). Search based software engineering. *Information and Software Technology*, 43(14), pp. 833–839.

6. Liubchenko, V. V. Decision support system for next release problem [Systema pidtrymky pryiniattia rishen dlia zadachi nastupnogo relizu]. *Shtuchnyi intelekt*, 1-2 (67-68), pp. 106–110.

7. Etgar, R., Gelbard, R. and Cohen, Yu. (2016). Project Scope Partitioning by Clustering Features into Releases of Long R&D Projects. *Procedia Computer Science*, 100, pp. 1235–1241.

## CLUSTERING APPLICATION FOR RELEASE PLANNING

**V. V. Liubchenko**
*Odessa National Polytechnic University*

***Abstract.*** *The software projects are characterized by a long planning horizon, which entails the policy of release management. A short-term version of this problem is known as the Next Release Problem. A central issue release planning is determining which features should be included in which releases. This problem is NP-hard and thus cannot be solved analytically. To reduce the complexity, it is proposed to apply a simple clustering algorithm. The similarity coefficient combines precedence based similarity, predecessor based similarity and successor based similarity. The resource constraints for the particular release define a clear cutting point for the cluster size. Proposed approach reduces the complexity of the problem from $O(n!)$, where n is the number of features, to $O(m^2)$, where m is the number of releases. One example is considered. There is pointed that to improve the results of features allocation to releases the priorities of features should be taken into account.*

***Key words:*** *release planning, next release problem, cauterization, similarity measure.*

## ПРРИМЕНЕНИЕ КЛАСТЕРИЗАЦИИ ДЛЯ ПЛАНИРОВАНИЯ РЕЛИЗОВ

**В. В. Любченко**
*Одеський національний політехнічний університет*

***Аннотация.*** *Рассмотрены задача планирования релизов и ее частный случай – задача следующего релиза. Предложены меры сходства для решения задачи планирования релизов с помощью методов кластеризации. Проанализированы результаты планирования релизов на основе кластеризации требований. Сделан вывод о необходимости введения дополнительных мер сходства.*

***Ключевые слова:*** *планирование релизов, задача следующего релиза, кластеризация, мера сходства.*

Отримано 11.04.2017

**Любченко Віра Вікторівна**, доктор технічних наук, доцент, професор кафедри системного програмного забезпечення Одеського національного політехнічного університету. Просп. Шевченка, 1, Одеса, Україна, E-mail: lvv@opu.ua, тел. +38-048-705-8566

**Vira Liubchenko**, Dr. of Science, Assoc. Professor, Professor at the Department of System Software, Odessa National Polytechnic University, Shevchenko ave., 1, Odessa, Ukraine

**ORCID ID:** 0000-0002-4611-7832