

УДК 004

DOI: 10.25140/2411-5363-2020-3(21)-209-219

Олександр Гордєєв, Костянтин Леонтієв

МОДЕЛЬ СЦЕНАРІЮ ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Актуальність теми дослідження. Оцінювання якості програмного забезпечення (ПЗ) є обов'язковим процесом при забезпеченні необхідної якості ПЗ у межах загального процесу його розробки. Постійний розвиток наявних і поява нових інформаційних технологій (штучний інтелект, хмарні обчислення, віртуальна й доповнена реальність тощо) і систем підвищує вимоги до процесу оцінювання та забезпечення якості ПЗ.

Постановка проблеми. Існуючі підходи до оцінювання якості мають суттєві недоліки, які пов'язані зі слабкою формалізацією при плануванні завдань оцінювання якості ПЗ, високим ступенем невизначеності при прийнятті рішення відповідальними учасниками процесу, недостатністю або надмірністю інформації, визначенням кількості учасників процесу оцінювання ПЗ.

Аналіз останніх досліджень і публікацій. Були розглянуті останні публікації, що знаходяться у відкритому доступі, у яких розглядається сценарний підхід у загальному вигляді й у частині оцінювання окремих завдань якості програмного забезпечення.

Виділення недосліджених частин загальної проблеми. Опис сценарного підходу для задач оцінювання якості програмного забезпечення.

Постановка завдання. Більш формалізоване представлення сценарного підходу для вирішення задач оцінювання якості ПЗ не розглядалося. Метою статті є розробка моделі сценарію оцінювання якості програмного забезпечення.

Виклад основного матеріалу. У статті пропонується представлення й опис моделі сценарію оцінювання якості ПЗ, яка складається з 6 наступних елементів: початкових умов, вхідних даних, дій, перехідних даних, ролей і результатів. Було встановлено, що протягом свого життєвого циклу сценарій може перебувати в наступних станах: сценарій на папері, пілотний сценарій і реальний сценарій. При переході в кожен стан множини елементів сценарію можуть змінюватися. Для формалізації таких змін були введені й формально описані додаткові операції над сценарієм: операція виключення та операція включення. Розглянуто варіанти нерівностей множин елементів сценарію для сценарію на папері й пілотного сценарію.

Висновки відповідно до статті. Представлена модель сценарію оцінювання якості програмного забезпечення може застосовуватися при оцінюванні якості ПЗ на основі засіву дефектів.

Ключові слова: сценарний підхід; модель сценарію; якість ПЗ; оцінювання якості ПЗ; засів дефектів ПЗ.

Рис.: 2. Табл.: 1. Бібл.: 16.

Актуальність теми дослідження. Оцінювання якості програмного забезпечення (ПЗ) є обов'язковим процесом при забезпеченні необхідної якості ПЗ у межах загального процесу його розробки. Постійний розвиток існуючих і поява нових інформаційних технологій (штучний інтелект, хмарні обчислення, віртуальна і доповнена реальність тощо) і систем підвищує вимоги до процесу оцінювання та забезпечення якості ПЗ загалом.

Постановка проблеми. Така еволюція серед існуючих підходів і парадигм оцінювання якості ПЗ має недостатню динаміку, оскільки мають місце недоліки, серед яких слабка формалізація при плануванні завдань оцінювання якості ПЗ, високий ступінь невизначеності при прийнятті рішення відповідальними учасниками процесу, недостатність або надмірність необхідної початкової інформації, формування групи учасників процесу оцінювання якості ПЗ. Особливо такі проблеми явно виражені в методах оцінювання якості ПЗ, заснованих на засіві дефектів ПЗ.

Аналіз останніх досліджень і публікацій. Одним із відомих підходів, який може до необхідного рівня формалізувати процес оцінювання якості ПЗ, є сценарний підхід. Відомі роботи, присвячені організації і формалізації процесу оцінювання якості ПЗ, описують окремі випадки в частині розвитку процесу оцінювання якості ПЗ [1-3], а сценарний підхід описується частково, на рівні деяких елементів [4-6]. Хоча існують роботи, присвячені сценарному підходу, але він розглядається в загальному вигляді як підхід до управління [7-10], без урахування особливостей оцінювання якості ПЗ загалом. Сценарний підхід концептуально не розглядається в роботах, присвячених оцінюванню якості ПЗ на основі засіву дефектів ПЗ [11; 12].

Виділення недосліджених частин загальної проблеми. Оскільки в існуючих роботах, які були проаналізовані в повному обсязі, не описується сценарний підхід до оцінювання якості програмного забезпечення, існує необхідність його формалізованого представлення.

Метою статті є розробка моделі сценарію оцінювання якості програмного забезпечення.

Виклад основного матеріалу.

Модель сценарію оцінювання якості програмного забезпечення. Поняття сценарію. Представимо і формально опишемо сценарієорієнтований підхід до оцінки якості ПЗ. Насамперед розглянемо поняття сценарію. Слово «сценарій» походить від латинського слова «*scaena*», яке перекладається як «сцена». Спочатку сценарій розглядався як літературно-драматичний твір, написаний як основа для постановки кіно- або телефільму, і інших заходів в театрі та інших місцях. У ХХ столітті провідний аналітик корпорації RAND Герман Кан (Herman Kahn) [13] адаптував цей термін для використання при написанні можливих історій розвитку подій у майбутньому. Олівер Сперроу (Oliver Sparrow), один з основоположників сценарного підходу в корпорації «Royal Dutch Shell», виділяє чотири сучасні трактування цього терміна [14]: «аналіз чутливості» в таких сферах як управління фінансовими потоками, оцінювання ризиків, менеджмент проєктів; синонім поняття «надзвичайний план» в військовому чи цивільному плануванні позаштатних ситуацій, визначаючи, хто і що повинен робити в разі виникнення нештатної ситуації; синонім «надзвичайного плану» в корпоративній або державній політиці; у розумінні «логічно узгоджене припущення про майбутнє» при прийнятті рішень і формуванні стратегії.

Усі основні визначення узагальнюються голландським ученим Філіпом Ван Ноттемом (Philip Van Notten) в [15] таке визначення: сценарій – це послідовний опис альтернативних гіпотетично можливих варіантів розвитку подій в майбутньому, який відображає різні погляди на минуле, сьогоднішнє та майбутнє, а також який може бути базисом для планування дій.

Формалізоване представлення моделі. Адаптувавши представлене визначення сценарію для оцінювання якості програмного забезпечення, отримаємо таке трактування: сценарій оцінювання якості ПЗ – це продукт планування та опису (безперервної) послідовності дій, спрямованих на оцінювання якості програмного забезпечення, який включає опис початкових умов, вхідних даних, очікуваного результату (гіпотези) і розподіл ролей учасників процесу оцінювання якості ПЗ. Серед ролей учасників процесу можна виділити такі: організатор (інженер-дослідник) процесу оцінювання якості ПЗ (розробник сценарію), керівник групи тестування ПЗ (керівник групи якості), тестувальник (інженер з якості), користувач. Таким чином, сценарій оцінювання якості програмного забезпечення включає 6 таких елементів: дії, перехідні дані, які передаються від етапу до етапу, ролі, вхідні дані, початкові умови, очікуваний результат або гіпотеза. Представимо в загальному вигляді – графічному (рис. 1) і формальному вигляді елементи сценарію:

– $INCONSCE = \{inconsce_k\}_{k=1}^p$ – множина початкових умов сценарію оцінювання якості програмного забезпечення ($INCONSCE$ – Initial Conditions of Scenario), $inconsce$ – початкова умова сценарію оцінювання якості програмного забезпечення;

– $INDASCE = \{indasce_j\}_{j=1}^f$ – множина вхідних даних сценарію оцінювання якості програмного забезпечення ($INDASCE$ – Input Data of Scenario), $indasce$ – вхідні дані сценарію оцінювання якості програмного забезпечення;

– $ACTSCE = \langle actsce_i \rangle_{i=1}^q$ – множина дій сценарію оцінювання якості програмного забезпечення ($ACTSCE$ – Actions of Scenario), $actsce$ – дія сценарію оцінювання якості програмного забезпечення;

– $TRANDAT = \{trandat_l\}_{l=1}^d$ – множина перехідних даних, які передаються від дії до дії, тобто вихідні дані переходять і стають вхідними даними для наступної дії ($TRANDAT$ – TRANSition DATA), $actsce$ – дія сценарію оцінювання якості програмного забезпечення;

- $ROLSCE = \{rolsce_n\}_{n=1}^t$ – множина ролей для сценарію оцінювання якості програмного забезпечення ($ROLSCE$ – Roles of Scenario), $rolsce$ – роль для сценарію оцінювання якості програмного забезпечення;
- $RESSCE = \{ressce_m\}_{m=1}^v$ – множина результатів сценарію оцінювання якості програмного забезпечення ($RESSCE$ – Results of Scenario), $ressce$ – результат сценарію оцінювання якості програмного забезпечення.

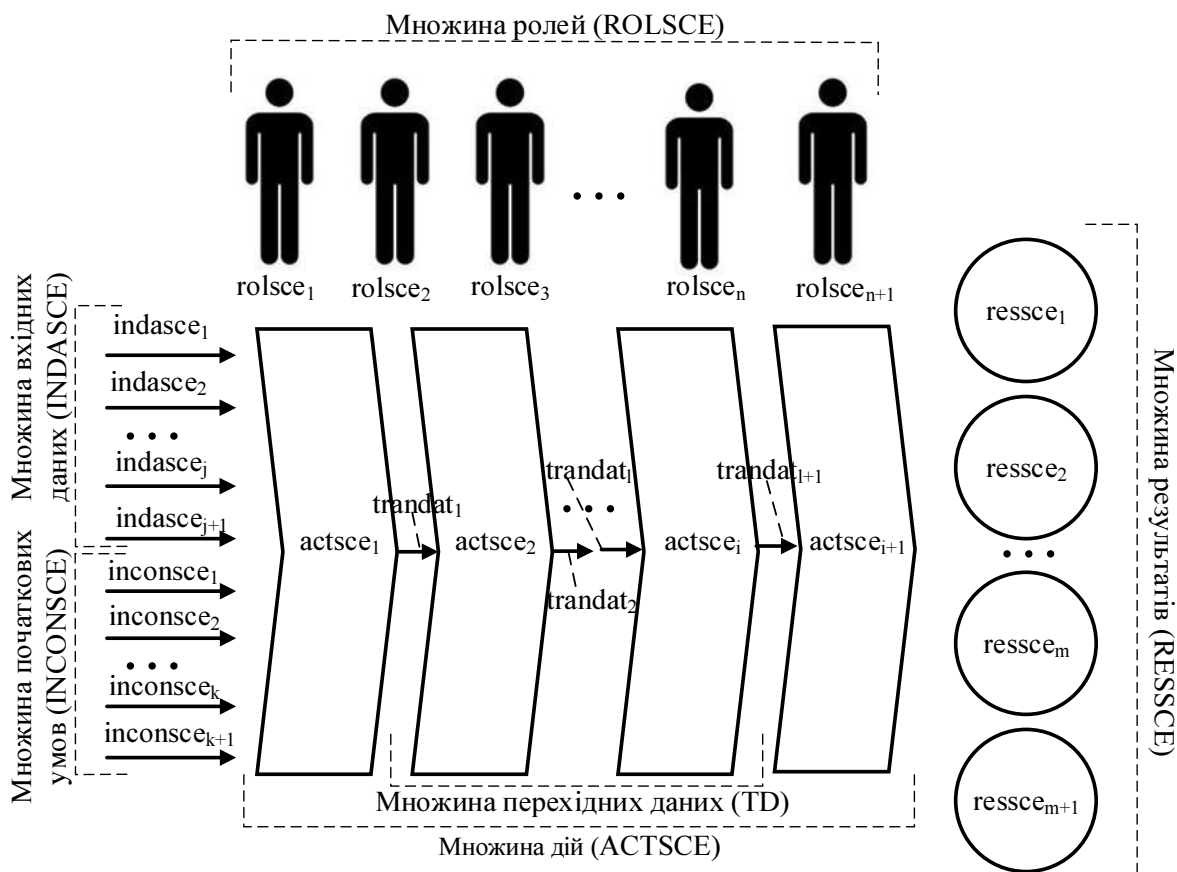


Рис. 1. Графічне представлення моделі сценарію оцінювання якості програмного забезпечення

Таким чином, сценарій оцінювання якості програмного забезпечення ($SAQSW$ – Scenario of Assessment of Quality of Software) описується як множина множин (1):

$$SAQSW = \left\{ \begin{array}{l} INCONCE, INDASCE, ACTSCE, \\ TRANDAT, ROLSCE, RESSCE \end{array} \right\} \quad (1)$$

Дослідним шляхом було встановлено, що сценарій протягом свого життєвого циклу (рис. 2) еволюціонує і представляється в трьох наступних станах:

- сценарій на папері ($SPAQSW$ – Scenario on Paper of Assessment of Quality of Software). Це перший стан сценарію, який розробляється організатором процесу оцінювання. Для позначення такого стану сценарію для кожної множини додано індекс «S» (2)

$$SPAQSW = \left\{ \begin{array}{l} INCONCE_s, INDASCE_s, ACTSCE_s, \\ TRANDAT_s, ROLSCE_s, RESSCE_s \end{array} \right\}; \quad (2)$$

– пілотний сценарій (*PSAQSW* – Pilot Scenario of Assessment of Quality of Software). Це сценарій на папері, який виконується в тестовому режимі. Такий сценарій необхідний, щоб відпрацювати й уточнити сценарій на реальному тестовому прикладі. Переважно кількість учасників, які залучені в сценарій, є мінімальною. Зазвичай такий сценарій відрізняється від сценарію на папері за рахунок уточнення елементів сценарію. Для позначення такого стану сценарію для кожної множини додано індекс «P» (3)

$$PSAQSW = \left\{ \begin{array}{l} INCONCE_p, INDASCE_p, ACTSCE_p, \\ TRANDAT_p, ROLSCE_p, RESSCE_p \end{array} \right\}; \quad (3)$$

– реальний сценарій (*RSAQSW* – Real Scenario of Assessment of Quality of Software). Такий стан сценарію використовується для оцінювання якості ПЗ для реального об'єкта дослідження. Здебільшого може відрізнятися від пілотного сценарію, за рахунок уточнень, які вносяться в нього в процесі виконання. Для позначення такого варіанта сценарію для кожної множини додано індекс «R» (4).

$$RSAQSW = \left\{ \begin{array}{l} INCONCE_R, INDASCE_R, ACTSCE_R, \\ TRANDAT_R, ROLSCE_R, RESSCE_R \end{array} \right\}. \quad (4)$$

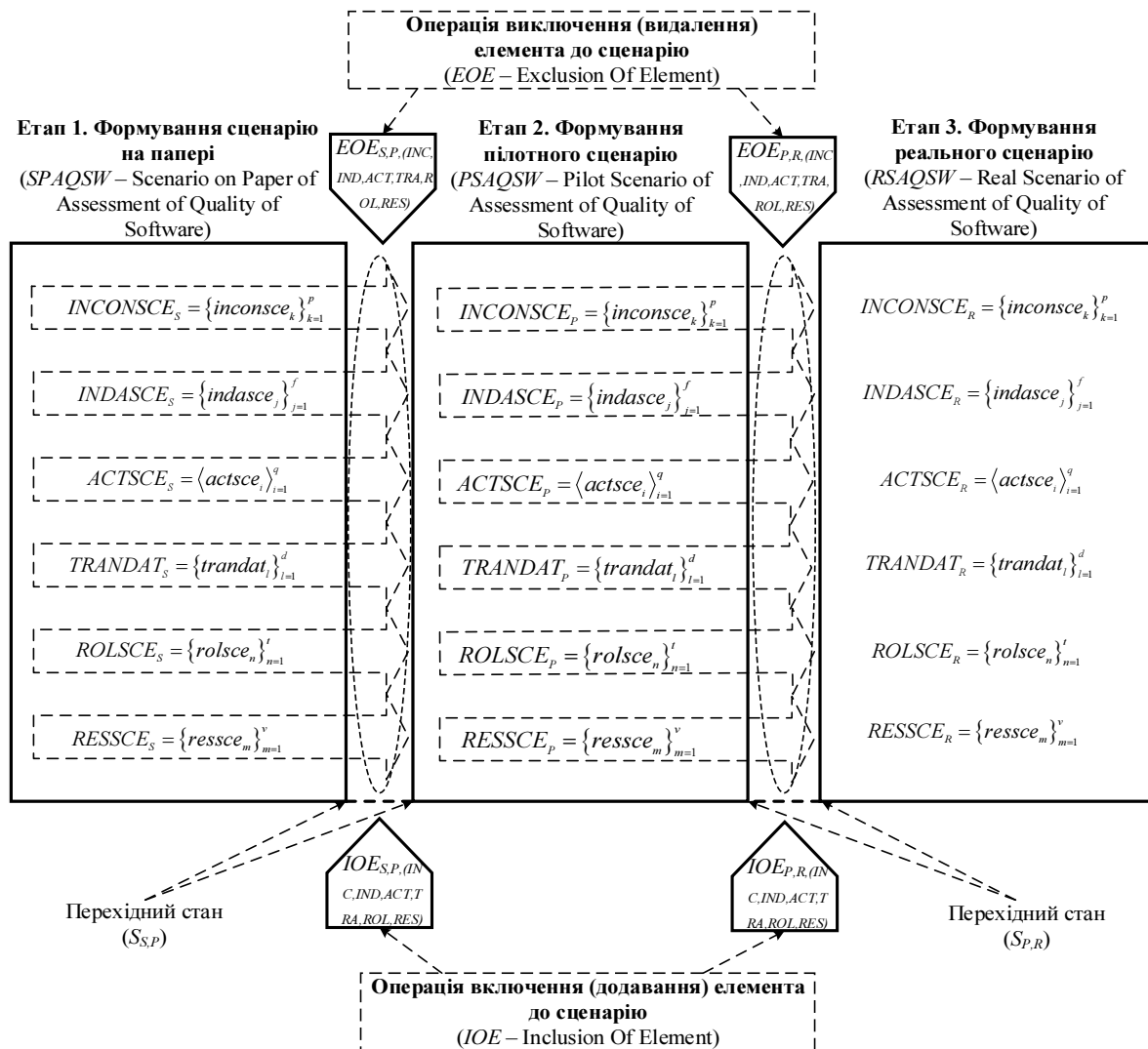


Рис. 2. Життєвий цикл сценарію оцінювання якості програмного забезпечення

Таким чином уточнимо загальний запис для сценарію оцінювання якості програмного забезпечення з урахуванням стану сценарію і додамо для кожної множини індекс VOS, який позначає стан сценарію (*VOS* – Variant Of Scenario). Отже, індекс VOS може набувати таких значень: S – *SPAQSW* – Scenario on Paper of Assessment of Quality of Software, P – *PSAQSW* – Pilot Scenario of Assessment of Quality of Software, R – *RSAQSW* – Real Scenario of Assessment of Quality of Software) (5):

$$SAQSW = \left\{ \begin{array}{l} INCONCE_{VOS}, INDASCE_{VOS}, ACTSCE_{VOS}, \\ TRANDAT_{VOS}, ROLSCE_{VOS}, RESSCE_{VOS} \end{array} \right\}. \quad (5)$$

Операції над сценарієм. Протягом свого життєвого циклу сценарій може уточнюватися, тобто видозмінюватися. У статті не розглядаються та не аналізуються приклади та причини, у яких випадках може змінюватися сценарій, оскільки такий матеріал вимагає більшого обсягу та може претендувати на окрему статтю. Було встановлено, що такі зміни зводяться до двох таких операцій над елементами сценарію:

- виключення (видалення) елемента сценарію (*EOE_{VOS,VOS,TEOS}* – Exclusion Of Element);
- включення (додавання) елемента сценарію (*IOE_{VOS,VOS,TEOS}* – Inclusion Of Element).

Можлива також операція перетворення елементів сценарію, але вона не розглядається, оскільки може бути представлена парою операцій виключення і включення. При введенні додаткових множин для кожної з них був доданий індекс *TEOS*, який позначає варіант сценарію (*TEOS* – Type of Elements Of Scenario). Таким чином, індекс *TEOS* може набувати таких значень: *INC* – *INCONSCE* – Initial Conditions of Scenario, *IND* – *INDASCE* – Input Data of Scenario, *ACT* – *ACTSCE* – Actions of Scenario, *TRA* – *TRANDAT* – TRANSITION DATA, *ROL* – *ROLSCE* – Roles of Scenario, *RES* – *RESSCE* – Results of Scenario).

Для більш формального опису таких змін сценарію введемо додаткове позначення – *S_{VOS,VOS}*, яке може бути двох видів: *S_{S,P}* – перехідний стан сценарію на папері в пілотний сценарій, *S_{P,R}* – перехідний стан пілотного сценарію в реальний сценарій. Розглянемо можливі варіанти нерівностей сценаріїв та їхніх елементів для перехідних станів (рис. 2).

Формально представимо опис операцій виключення (*EOE_{VOS,VOS,TEOS}*) і включення (*IOE_{VOS,VOS,TEOS}*). Для цього необхідно ввести додаткові наступні множини:

- множина початкових елементів з відповідної множини (*SOE_{TEOS}* – Set of Original Elements). Така множина включає всі елементи початкового сценарію, для якого буде застосовуватися відповідна операція;

- множина елементів, що виключаються з відповідної множини (*SEXE_{TEOS}* – Set of Excluding Elements). Оскільки при використанні операції виключення може видалитися тільки один елемент із множини початкових елементів сценарію, то така множина включатиме один елемент, хоча такі елементи в множині можуть накопичуватися при багаторазовому використанні операції виключення для початкового сценарію;

- множина виключених елементів з відповідної множини (*SEE_{TEOS}* – Set of Excluded Elements). Оскільки при використанні операції виключення може видалитися тільки один елемент з множини початкових елементів сценарію, то така множина буде складатися з одного елементу, хоча такі елементи в множині можуть накопичуватися при багаторазовому використанні операції виключення для початкового сценарію. Тобто елемент сценарію при використанні операції виключення переходить з множини елементів, що виключаються, у множину виключених елементів;

- множина результуючих елементів з відповідної множини (*SRE_{TEOS}* – Set of Resulting Elements). Така множина формується як різниця між множиною початкових елементів і множиною елементів сценарію, що виключаються, або ж як сума множини початкових елементів і множини елементів, що виключаються;

– множина елементів, що включаються, з відповідної множини (SIE_{TEOS} – Set of Included Elements). Це множина, яка включає в себе елемент або елементи, які будуть додані в множину початкових елементів, і таке об'єднання множин формує результуючу множину елементів сценарію.

У загальному вигляді операція виключення (видалення) елемента ($EOE_{VOS,VOS,TEOS}$) для кожного стану сценарію записується в такий спосіб (6):

$$EOE_{VOS,VOS,TEOS} = \left\{ \begin{array}{l} SRE_{TEOS} = SOE_{TEOS} \setminus SEXE_{TEOS} \\ SEE_{TEOS} = SOE_{TEOS} \cap SEXE_{TEOS} \\ SIE_{TEOS} = \emptyset \end{array} \right\}, \quad (6)$$

а операція включення (додавання) елемента ($IOE_{VOS,VOS,TEOS}$) для кожного стану сценарію таким чином (7):

$$IOE_{VOS,VOS,TEOS} = \left\{ \begin{array}{l} SRE_{TEOS} = SOE_{TEOS} \cup SIE_{TEOS} \\ SOE_{TEOS} \cap SIE_{TEOS} = \emptyset \\ SEE_{TEOS} = SEXE_{TEOS} = \emptyset \end{array} \right\}. \quad (7)$$

Життєвий цикл сценарію оцінювання якості програмного забезпечення включає 3 етапи, які відповідають його трьом станам (див. рис. 2):

1. Етап формування сценарію на папері;
2. Етап формування пілотного сценарію;
3. Етап формування реального сценарію.

Такі етапи формування сценарію оцінювання якості програмного забезпечення можуть виконуватися тільки послідовно. На початку – етап формування сценарію на папері, далі пілотного та реального сценаріїв.

Переходячи від етапу до етапу, протягом свого життєвого циклу сценарій оцінювання якості програмного забезпечення може зазнавати певних змін. Такі зміни є наслідком застосування операцій виключення і (або) включення елементів сценарію.

Для перехідного стану $S_{S,P}$ існують наступні два варіанти нерівностей:

– сценарій на папері не дорівнює пілотному сценарію, тобто $SPAQSW \neq PSAQSW$.

Якщо розглядати таку нерівність на рівні елементів, то можливі варіанти рівності і нерівності таких елементів. Розглянемо варіанти нерівностей на рівні елементів сценарію на папері і пілотного сценарію і представимо їх у вигляді повного простого перебору, включаючи варіант повної рівності (див. табл.). Для прикладу один такий варіант нерівностей (таблиця, рядок 22) розпишемо більш детально. Такий варіант складається з таких співвідношень: $INCONCE_s \neq INCONCE_p$, $INDASCE_s = INDASCE_p$, $ACTSCE_s \neq ACTSCE_p$, $TRANDT_s = TRANDT_p$, $ROLSCE_s \neq ROLSCE_p$, $RESSCE_s = RESSCE_p$. Оскільки серед нерівностей є і рівності, які вказують на ідентичність елементів, будемо розглядати і описувати тільки наступні нерівності: $INCONCE_s \neq INCONCE_p$, $ACTSCE_s \neq ACTSCE_p$, $ROLSCE_s \neq ROLSCE_p$. Оскільки нерівності сценаріїв свідчать про використання операції включення або виключення опишемо більш детально такі нерівності для обох операцій:

– якщо застосували операцію виключення для елементів множини початкових умов, дій і ролей для сценарію (8-10):

$$EOE_{S,P,INC} = \left\{ \begin{array}{l} SOE_{INC} = INCONCE_S \\ SRE_{INC} = SOE_{INC} \setminus SEXE_{INC} = INCONCE_P \\ SEE_{INC} = SOE_{INC} \cap SEXE_{INC} \\ SIE_{INC} = \emptyset \end{array} \right\}, \quad (8)$$

Таблиця

Варіанти нерівностей множин елементів для сценарію на папері і пілотного сценарію

№	$INCONCE_{S(P)},$ $INCONCE_{P(R)}$	$INDASCE_{S(P)},$ $INDASCE_{P(R)}$	$ACTSCE_{S(P)},$ $ACTSCE_{P(R)}$	$TRANDAT_{S(P)},$ $TRANDAT_{P(R)}$	$ROLSCE_{S(P)},$ $ROLSCE_{P(R)}$	$RESSCE_{S(P)},$ $RESSCE_{P(R)}$
1	2	3	4	5	6	7
1.	≠	≠	≠	≠	≠	≠
2.	≠	≠	≠	≠	≠	=
3.	≠	≠	≠	≠	=	≠
4.	≠	≠	≠	≠	=	=
5.	≠	≠	≠	=	≠	≠
6.	≠	≠	≠	=	≠	=
7.	≠	≠	≠	=	=	≠
8.	≠	≠	≠	=	=	=
9.	≠	≠	=	≠	≠	≠
10.	≠	≠	=	≠	≠	=
11.	≠	≠	=	≠	=	≠
12.	≠	≠	=	≠	=	=
13.	≠	≠	=	=	≠	≠
14.	≠	≠	=	=	≠	=
15.	≠	≠	=	=	=	≠
16.	≠	≠	=	=	=	=
17.	≠	=	≠	≠	≠	≠
18.	≠	=	≠	≠	≠	=
19.	≠	=	≠	≠	=	≠
20.	≠	=	≠	≠	=	=
21.	≠	=	≠	=	≠	≠
22.	≠	=	≠	=	≠	=
23.	≠	=	≠	=	=	≠
24.	≠	=	≠	=	=	=
25.	≠	=	=	≠	≠	≠
26.	≠	=	=	≠	≠	=
27.	≠	=	=	≠	=	≠
28.	≠	=	=	≠	=	=
29.	≠	=	=	=	≠	≠
30.	≠	=	=	=	≠	=
31.	≠	=	=	=	=	≠
32.	≠	=	=	=	=	=
33.	=	≠	≠	≠	≠	≠
34.	=	≠	≠	≠	≠	=
35.	=	≠	≠	≠	=	≠
36.	=	≠	≠	≠	=	=
37.	=	≠	≠	=	≠	≠
38.	=	≠	≠	=	≠	=
39.	=	≠	≠	=	=	≠
40.	=	≠	≠	=	=	=
41.	=	≠	=	≠	≠	≠
42.	=	≠	=	≠	≠	=
43.	=	≠	=	≠	=	≠
44.	=	≠	=	≠	=	=

Закінчення табл.

1	2	3	4	5	6	7
45.	=	≠	=	=	≠	≠
46.	=	≠	=	=	≠	=
47.	=	≠	=	=	=	≠
48.	=	≠	=	=	=	=
49.	=	=	≠	≠	≠	≠
50.	=	=	≠	≠	≠	=
51.	=	=	≠	≠	=	≠
52.	=	=	≠	≠	=	=
53.	=	=	≠	=	≠	≠
54.	=	=	≠	=	≠	=
55.	=	=	≠	=	=	≠
56.	=	=	≠	=	=	=
57.	=	=	=	≠	≠	≠
58.	=	=	=	≠	≠	=
59.	=	=	=	≠	=	≠
60.	=	=	=	≠	=	=
61.	=	=	=	=	≠	≠
62.	=	=	=	=	≠	=
63.	=	=	=	=	=	≠

$$EOE_{S,P,IND} = \left\{ \begin{array}{l} SOE_{IND} = INDASCE_S \\ SRE_{IND} = SOE_{IND} \setminus SEXE_{IND} = INDASCE_P \\ SEE_{IND} = SOE_{IND} \cap SEXE_{IND} \\ SIE_{IND} = \emptyset \end{array} \right\}, \quad (9)$$

$$EOE_{S,P,ROL} = \left\{ \begin{array}{l} SOE_{ROL} = ROLSCE_S \\ SRE_{ROL} = SOE_{ROL} \setminus SEXE_{ROL} = ROLSCE_P \\ SEE_{ROL} = SOE_{ROL} \cap SEXE_{ROL} \\ SIE_{ROL} = \emptyset \end{array} \right\}; \quad (10)$$

– якщо застосували операцію включення для елементів множини початкових умов, дій і ролей для сценарію (11-13):

$$IOE_{S,P,INC} = \left\{ \begin{array}{l} SOE_{INC} = INCONCE_S \\ SRE_{INC} = SOE_{INC} \cup SEXE_{INC} = INCONCE_P \\ SOE_{INC} \cap SIE_{INC} = \emptyset \\ SEE_{INC} = SEXE_{INC} = \emptyset \\ SIE_{INC} = \emptyset \end{array} \right\}, \quad (11)$$

$$IOE_{S,P,IND} = \left\{ \begin{array}{l} SOE_{IND} = INDASCE_S \\ SRE_{IND} = SOE_{IND} \cup SEXE_{IND} = INDASCE_P \\ SOE_{IND} \cap SIE_{IND} = \emptyset \\ SEE_{IND} = SEXE_{IND} = \emptyset \\ SIE_{IND} = \emptyset \end{array} \right\}, \quad (12)$$

$$IOE_{S,P,ROL} = \left\{ \begin{array}{l} SOE_{ROL} = ROLSCE_S \\ SRE_{ROL} = SOE_{ROL} \cup SEXE_{ROL} = ROLSCE_P \\ SOE_{ROL} \cap SIE_{ROL} = \emptyset \\ SEE_{ROL} = SEXE_{ROL} = \emptyset \\ SIE_{ROL} = \emptyset \end{array} \right\}. \quad (13)$$

– сценарій на папері ідентичний пілотному сценарію, тобто $SPAQSW = PSAQSW$. Таким чином кожній множині сценарію на папері відповідає еквівалентна множина пілотного сценарію, тобто $INCONSCE_S = INCONSCE_P$, $INDASCE_S = INDASCE_P$, $ACTSCE_S = ACTSCE_P$, $TRANDAT_S = TRANDAT_P$, $ROLSCE_S = ROLSCE_P$.

Для перехідного стану $S_{P,R}$ існують наступні два варіанти нерівностей:

1. Пілотний сценарій не дорівнює реальному сценарію, тобто $PSAQSW \neq RSAQSW$. Така нерівність в частині множини варіантів на рівні елементів сценарію аналогічна нерівності сценарію на папері й пілотного сценарію з урахуванням того, що як коефіцієнти при елементах сценарію розглядаються коефіцієнти в круглих дужках, тобто замість індексу «S» розглядається індекс «P», а замість індексу «P» розглядається індекс «R» (див. табл.).

2. Пілотний сценарій ідентичний реальному сценарію, тобто $PSAQSW = RSAQSW$. Таким чином кожна з множин елементів одного сценарію дорівнює відповідній множині іншого сценарію, тобто $INCONSCE_P = INCONSCE_R$, $INDASCE_P = INDASCE_R$, $ACTSCE_P = ACTSCE_R$, $TRANDAT_P = TRANDAT_R$, $ROLSCE_P = ROLSCE_R$.

Застосування моделі. Запропонована модель може бути використана для оцінювання якості програмного забезпечення з використанням засіву дефектів ПЗ. Зокрема, при розробці та реалізації процедур засіву дефектів (Fault Injection Testing, FIT), які використовуються в Науково-виробничому підприємстві «Радій», застосовуються різні сценарії засіву для оцінки функціональної безпеки FPGA проектів для локальних інформаційно-керуючих систем АЕС. Для виконання FIT формуються різні профілі дефектів, що засіваються в електронний проєкт, фізичний модуль, програмне забезпечення верхнього рівня, одиночних і множинних дефектів, відносних і абсолютних дефектів двоверсійних систем аварійного захисту тощо. Таке різноманіття профілів породжує різноманіття сценаріїв FIT і сценаріїв оцінювання якості, описаних у статті.

Висновки відповідно до статті. У статті представлена і формально описана модель сценарію оцінювання якості програмного забезпечення. Її застосування дозволить формалізувати планування (початкові умови, вхідні дані, дії, перехідні дані, ролі і результати) і виконання сценарію, врахувати можливі особливості станів сценаріїв, перехід сценарію зі стану в стан з урахуванням можливих змін множин елементів сценарію.

Подальші дослідження доцільно спрямувати на розробку й автоматизацію реалізації деталізованих сценаріїв оцінювання якості ПЗ і FPGA проєктів.

Список використаних джерел

1. Ratanothayanon S., Srivisal Ch., Vanichayobonand S., Preechaveerakul L. Comparative Classifiers for Software Quality Assessment. 2012. *International Journal of Engineering and Technology*. 2012. Vol. 4(4). Pp. 404–408.
2. Gordieiev O., Kharchenko V. IT-oriented software quality models and evolution of the prevailing characteristics. In *proceedings of IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2018. Pp. 390–395.

3. Brandtner M. Fostering software quality assessment. *In proceedings of 35th International Conference on Software Engineering (ICSE)*. 2013. Pp. 1393–1396.
4. Software quality models: purposes, usage scenarios and requirements / Deissenboeck F., Juergens E., Lochmann K. et al. *In proceedings of the ICSE Workshop on Software Quality*. 2009. Pp. 9-14.
5. Briggs Ch. M., Matejova M. Scenario Planning and Complex Scenario Approach. *Disaster Security*. 2019. Pp. 38-60.
6. Koppen P. J., Mackor A. R. A Scenario Approach to the Simonshaven Case. *Wiley Online Library. Topics in Cognitive Science*. 2019. Pp. 1–20. URL: <https://onlinelibrary.wiley.com/doi/10.1111/tops.12429>.
7. Campi M. C., Garatti S. Introduction to the Scenario Approach : book. *Society for Industrial and Applied Mathematics and the Mathematical Optimization Society*. 2018. 116 p.
8. Ramponi F. A. Consistency of the Scenario Approach. *Journal on Optimization*. 2018. Vol. 28(1). Pp. 135–162.
9. Garatti S., Campi M. C. Learning for Control: a Bayesian Scenario Approach. *In proceedings of the IEEE 58th Conference on Decision and Control (CDC)*. 2019. Pp. 1772–1777.
10. Shumkov Y. A., Vidovskiy L. A. Scenario approach to project management. *Polythematic Online Scientific Journal of Kuban State Agrarian University*. 2017. Vol. 134 (10). Pp. 1-9.
11. Soumya S., Baiju A. Software Faults Emulation by Software Fault Injection. *International Journal of Computer Applications*. 2014. Vol. 97. Pp. 9-11.
12. Cotroneo D., Madeira H. Introduction to Software Fault Injection. *Innovative Technologies for Dependable OTS-Based Critical Systems / ed. D. Cotroneo*. Springer. 2013. Pp. 1-15.
13. Kahn H. The Year 2000: A framework for speculation on the next thirty-three year. *NY: Macmillan Publishing Company*. 1967. 432 p.
14. Sparrow O. Making use of scenarios – from the vague to the concrete. *Scenario & Strategy Planning*. 2000. Vol. 2(5). Pp. 18–21.
15. Van Notten Ph. Writing on the wall: scenario development in times of discontinuity. *Florida: Boca Raton*. 2005. 209 p.
16. Kharchenko V., Gordieiev O., Vilkomir S., Odarushchenko O. T-Wise-Based Multi-Fault Injection Technique for the Verification of Safety Critical I&C Systems. *In proceedings of the 9th International Conference on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT)*. 2015. P.1-9.

References

1. Ratanothayanon, S., Srivisal, Ch., Vanichayobonand, S., Preechaveerakul, L. (2012). Comparative Classifiers for Software Quality Assessment. *International Journal of Engineering and Technology*, 4(4), pp. 404–408.
2. Gordieiev, O., Kharchenko, V. (2018). IT-oriented software quality models and evolution of the prevailing characteristics. *In proceedings of IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT'2018)* (pp. 390–395).
3. Brandtner, M. (2013) Fostering software quality assessment. *In proceedings of 35th International Conference on Software Engineering (ICSE'2013)*. (pp. 1393–1396).
4. Deissenboeck, F., Juergens, E., Lochmann, K., et al. (2009). Software quality models: purposes, usage scenarios and requirements. *In proceedings of the ICSE Workshop on Software Quality* (pp. 9–14).
5. Briggs, Ch. M., Matejova, M. (2019). Scenario Planning and Complex Scenario Approach. *Disaster Security* (pp. 38–60).
6. Koppen, P. J., Mackor, A. R. (2019). A Scenario Approach to the Simonshaven Case. *Wiley Online Library. Topics in Cognitive Science* (pp. 1–20). <https://onlinelibrary.wiley.com/doi/10.1111/tops.12429>.
7. Campi, M. C., Garatti, S. (2018). *Introduction to the Scenario Approach* : book. Society for Industrial and Applied Mathematics and the Mathematical Optimization Society. 116 p.
8. Ramponi, F. A. (2018). Consistency of the Scenario Approach. *Journal on Optimization*, 28(1), pp. 135–162.
9. Garatti, S., Campi, M. C. (2019). Learning for Control: a Bayesian Scenario Approach. *In proceedings of the IEEE 58th Conference on Decision and Control (CDC'2019)*. (pp. 1772–1777).

10. Shumkov, Y. A., Vidovskiy, L. A. (2017). Scenario approach to project management. *Polythematic Online Scientific Journal of Kuban State Agrarian University*, 34 (10), pp. 1–9.
11. Soumya, S., Baiju, A. (2014). Software Faults Emulation by Software Fault Injection. *International Journal of Computer Applications*, 97, pp. 9–11.
12. Cotroneo, D., Madeira, H. (2013). Introduction to Software Fault Injection. *Innovative Technologies for Dependable OTS-Based Critical Systems* / ed. D. Cotroneo. Springer (pp. 1–15).
13. Kahn, H. (1967). *The Year 2000: A framework for speculation on the next thirty-three year*. Macmillan Publishing Company.
14. Sparrow, O. (2000). Making use of scenarios – from the vague to the concrete. *Scenario & Strategy Planning*, 2(5), pp. 18–21.
15. Van Notten, Ph. (2005). *Writing on the wall: scenario development in times of discontinuity*. Florida: Boca Raton.
16. Kharchenko, V., Gordieiev, O., Vilkomir, S., Odarushchenko, O. (2015). T-Wise-Based Multi-Fault Injection Technique for the Verification of Safety Critical I&C Systems. *In proceedings of the 9th International Conference on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT'2015)* (pp. 1–9).

UDC 004

Oleksandr Gordieiev, Konstantin Leontiev

SOFTWARE QUALITY ASSESSMENT SCENARIO MODEL

Urgency of the research. Assessment of the software quality is a mandatory process to ensure the required quality of software within the overall software development process. The constant development of existing and the emergence of new information technologies (artificial intelligence, cloud computing, virtual and augmented reality, etc.) and systems increases the requirements for the evaluation process and software quality assurance.

Target setting. Existing approaches to quality assessment have significant problems associated with weak formalization in planning software quality assessment tasks, a high degree of uncertainty in decision making by the responsible participants in the process, insufficient or redundant information, and determining the number of participants in the software assessment process.

Actual scientific researches and issues analysis. Recent publications in open access, which consider the scenario approach in general and in relation to the tasks of assessing the quality of software, were analyzed.

Uninvestigated parts of general matters defining. Description of the scenario approach for software quality assessment tasks.

The research objective. A more formalized presentation of the scenario approach for solving tasks of software quality assessment has not been considered. The purpose of the article is to develop a model of the software quality assessment scenario.

The statement of basic materials. The article proposes a representation and description of the software quality assessment scenario model, which consists of the following 6 elements: initial conditions, input data, actions, transition data, roles and results. It has been found, that during its life cycle, a scenario can be in the following states: scenario on paper, pilot scenario and real scenario. During the transition to each state, many elements of the script can change. To formalize such changes, additional operations on the script were introduced and formally described: the operation of exclusion and the operation of inclusion. Variants of set inequalities for the scenario on paper and the pilot scenario are considered.

Conclusions. The model of the software quality assessment scenario is presented. It can be used for software quality assessment based on the seeding of defects.

Keywords: scenario approach; scenario model; software quality; software quality assessment; software defects injection.

Fig.: 2. Table: 1. References: 16.

Гордєєв Олександр Олександрович – кандидат технічних наук, доцент, завідувач кафедри кібербезпеки, Університет банківської справи (вул. Андріївська, 1, м. Київ, 04070, Україна).

Gordieiev Oleksandr – PhD in Technical Sciences, Associate professor, Head of cybersecurity department, Banking university (1 Andriivska Str., 04070 Kyiv, Ukraine).

E-mail: alex.gordeyev@gmail.com

Леонтєв Костянтин Петрович – технічний директор, Науково-виробниче підприємство «Радій» (вул. Академіка Тамма, 29, г. Кропивницький, 25009, Україна).

Leontiev Konstantin – Technical Director, Research and Production Enterprise “Radiy” (29 Akademika Tamma Str., 25009 Kropyvnytskyi, Ukraine).