тально розглянуто архітектуру баз знань для моделювання систем, що описуються лінійними інтегральними рівняннями. Розглянутий система, заснована на знаннях у вигляді композиції конкретної функціональної мережі та експертної системи.

Ключові слова: *моделювання, динамічні системи, інтелектуальне програмне забезпечення, інтегральні рівняння, база знань, функціональна мережа, експертна система*.

Отримано: 3.09.2020

УДК 004
DOI: 10.32626/2308-5916.2020-21.51-60

О. О. Гордєєв*, канд. техн. наук,
К. П. Леонтієв**

* Університет банківської справи, м. Київ,
** Науково-виробниче підприємство «Радій», м. Кропивницький

МОДЕЛЬ ЖИТТЄВОГО ЦИКЛУ ДЕФЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Процес розробки програмного забезпечення включає в себе обов'язковий додатковий процес забезпечення якості програмного забезпечення, який являє собою сукупність заходів, що охоплюють всі технологічні етапи розробки, випуску та експлуатації програмного забезпечення інформаційних систем, що проводяться на різних етапах життєвого циклу програмного забезпечення для забезпечення необхідного рівня якості програмного забезпечення. Одне з основних завдань такого процесу полягає в знаходженні і усуненні дефектів програмного забезпечення. Дана робота присвячена формальному представленню життєвого циклу дефекту програмного забезпечення. Модель життєвого циклу дефекту програмного забезпечення розглядається як ланцюжок, який починається з помилки розробника і закінчується відмовою програмного забезпечення. У статті подається загальна структура моделі дефекту життєвого циклу програмного забезпечення, яка включає в себе помилку розробника, помилку оператора, прихований дефект у програмному забезпеченні, активний дефект у програмному забезпеченні, помилку обчислення, збій або відмову, породжену вразливість, активовану вразливість, несанкціоноване управління та несанкціонований доступ до даних. Така модель деталізується в набір патологічних ланцюжків, які структурно представляють модифікації життєвого циклу дефекту програмного забезпечення з урахуванням природи виникнення самого дефекту програмного забезпечення. Серед патологічних ланцюжків виділяють наступні: фізичний, проектування, розробки та взаємодії.

Патологічний ланцюжок взаємодії деталізується ще на патологічний ланцюжок взаємодії внаслідок фізичного впливу і патологічний ланцюжок взаємодії внаслідок інформаційного впливу. Дана модель життєвого циклу дефекту програмного забезпечення проектується на V-подібну модель розробки програмного забезпечення, що дозволяє представити своєрідну еволюцію дефекту програмного забезпечення на кожному етапі життєвого циклу розробки програмного забезпечення окремо і в рамках всієї V-подібної моделі розробки програмного забезпечення.

Ключові слова: *якість програмного забезпечення, дефект програмного забезпечення, життєвий цикл дефекту програмного забезпечення, життєвий цикл програмного забезпечення, патологічні ланцюжки.*

Вступ. Розвиток існуючих і поява нових інформаційних технологій призводять до ускладнення життєвого циклу розробки програмного забезпечення. Чим складніші інформаційні технології, тим більш складним стає процес розробки програмного забезпечення (ПЗ). Така динаміка передбачає збільшення числа дефектів при розробці ПЗ. Дефект ПЗ — це результат, який має певні причини і наслідки. Причинами, як правило, є помилки розробників, а наслідки можуть виражатися в збоях або відмовах інформаційних систем.

Постановка задачі. Дефекти ПЗ аналізуються і досліджуються з моменту появи програмної інженерії як окремого інженерного напрямку [1]. З того часу викристалізувався і успішно розвивається в рамках програмної інженерії напрям забезпечення якості ПЗ. Було розроблено багато підходів, методів, методик і інструментальних засобів, спрямованих на забезпечення якості ПЗ. Не дивлячись на це, проблеми в частині появи дефектів ПЗ актуальні і сьогодні [2-4]. На думку авторів статті, у рамках дослідження дефектів ПЗ доцільно більш детально дослідити сам дефект ПЗ і описати його життєвий цикл в рамках процесу розробки ПЗ. Існуючі роботи в повній мірі не розглядають детально дефект ПЗ [5-6], у них взагалі не досліджені або не розглянуті не в повній мірі причинно-наслідкові зв'язки появи дефектів ПЗ [7-8], життєвий цикл (ЖЦ) дефекту ПЗ, а також його прив'язка до життєвого циклу розробки ПЗ [9-10]. У зв'язку з цим метою статті є розробка моделі життєвого циклу дефекту ПЗ і його проекція на ЖЦ розробки ПЗ.

Життєвий цикл дефекту ПЗ. Життєвий цикл дефекту ПЗ буде розглядатися на основі наступної базової послідовності «помилка розробника — відмова ПЗ»: помилка розробника (оператора) (mistake), дефект в програмному забезпеченні (defect або bug), помилка обчислення при роботі програмного забезпечення (error), збій (fault), відмова (failure) (рис. 1).

Спираючись на послідовність «помилка розробника — відмова ПЗ» сформуємо базову структуру життєвого циклу дефекту ПЗ. Така струк-

тура включає в себе наступні основні компоненти: причини, результат, наслідки та побічні наслідки (рис. 1). Перш за все, для виникнення результату — дефекту (defect, D) і помилки обчислення (error, E) повинна передувати причина або джерело, як правило, це помилки розробників або оператора (mistake, M). Дефект в програмному забезпеченні приводить до помилки обчислення. Наслідками помилки обчислення є збій (fault, F) або відмова (failure, FR). Варто також відзначити, що дефекти ПЗ можуть привести до виникнення уразливості (vulnerability, V), через яку зловмисники можуть отримувати несанкціонований доступ (unauthorized access, UA) до програмного забезпечення і в підсумку приводити до збою або відмови (рис. 2).

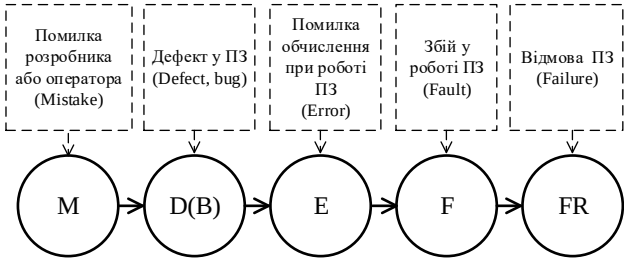

Рис. 1. *Послідовність «помилка розробника — відмова ПЗ»*

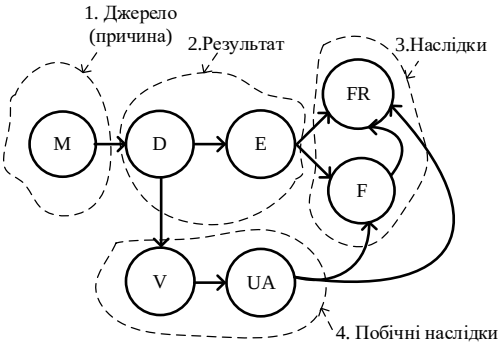

Рис. 2. *Загальна структура моделі ЖЦ дефекту ПЗ*

Таким чином, множина елементів життєвого циклу дефекту ПЗ (Software Defect Life Cycle, SDLC) складається з 7 елементів і матиме наступний вигляд (1):

$$SDLC = \{M, D, E, FR, F, V, UA\}. \quad (1)$$

Більш детальний аналіз ЖЦ ПЗ дозволив його деталізувати і сформувати так звані модифікації ЖЦ ПЗ, які отримали назву «патологічні ланцюжки». Патологічні ланцюжки — це послідовність взаємопов'язаних подій, які можуть виникнути при розробці та використанні програм-

ного забезпечення людино-комп'ютерних систем, починаючи від помилки програміста (оператора) і закінчуючи відмовою інформаційної системи в цілому. Як правило, всі такі патологічні ланцюжки мають уніфіковану структуру, а відрізняються, перш за все, природою джерела виникнення ланцюжка (причиною). Розглянемо структуру патологічного ланцюжка. Вона включає в себе наступні елементи: помилку розробника (development mistake, DM), помилку оператора (operator mistake, OM), прихований дефект у програмному забезпеченні (hidden defect, HD), активний дефект в програмному забезпеченні (active defect, AD), помилку обчислення (error, E), збій (fault, F), відмову (failure, FR), породжену вразливість (created vulnerability, CV), активовану вразливість (activated vulnerability, AV), несанкціоноване управління (unauthorized control, UC), несанкціонований доступ до даних (unauthorized access, UA) (рис. 3).

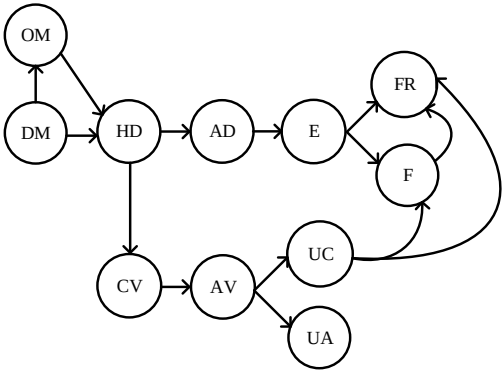

***Рис. 3**. Структура патологічного ланцюжка*

Таким чином, множина елементів патологічного ланцюжка, модифікації життєвого циклу дефекту ПЗ, розширилася до 11 елементів і матиме вже наступний уточнений вид (2):

$$SDLC = \{OM, DM, HD, AD, E, FR, F, CV, AV, UC, UA\}. \qquad (2)$$

Серед патологічних ланцюжків можна виділити фізичний, проектування, розробки та взаємодії. Розглянемо їх більш детально:

1. Фізичний патологічний ланцюжок. Природа виникнення джерела цього ланцюжка є фізичною, тобто це фізичні дефекти або несправності апаратного забезпечення. Елементи такого ланцюжка будемо позначати індексом — буквою p (physical) (рис. 4). Множина елементів такого ланцюжка практично ідентична множині елементів уніфікованої структурі. У фізичного патологічного ланцюжка додається новий елемент — знос апаратного забезпечення (hardware wear, HR). Вважається, що знос апаратного забезпечення є також джерелом дефектів програмного забезпечення (рис. 4).

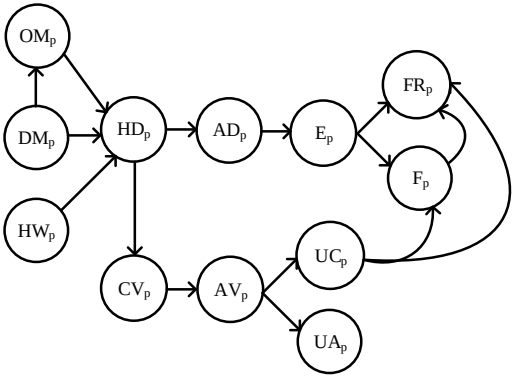

Рис. 4. *Фізичний патологічний ланцюжок*

Відзначимо, що множина елементів патологічного ланцюжка, модифікації життєвого циклу дефекту ПЗ, збільшилася до 12 елементів і матиме вже наступний уточнений вид (2):

$$SDLC_p = \\ = \left\{OM_p, DM_p, HD_p, AD_p, E_p, FR_p, F_p, HW_p, CV_p, AV_p, UC_p, UA_p\right\}. \quad (2)$$

2. Патологічний ланцюжок проектування. Природа виникнення джерела цього ланцюжка полягає в процесі проектування, тобто в дефектах проектування. Елементи такого ланцюжка будемо позначати індексом — буквами des (design) (рис. 5). Множина елементів патологічного ланцюжка проектування, модифікації життєвого циклу дефекту ПЗ, матиме наступний уточнений вид (3):

$$SDLC_{des} = \begin{Bmatrix} OM_{des}, DM_{des}, HD_{des}, AD_{des}, E_{des}, \\ FR_{des}, F_{des}, CV_{des}, AV_{des}, UC_{des}, UA_{des} \end{Bmatrix}. \quad (3)$$

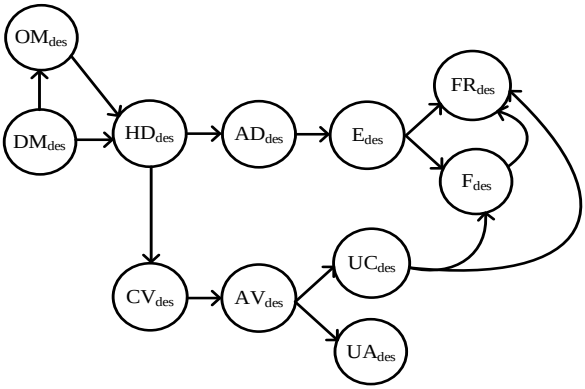

Рис. 5. *Патологічний ланцюжок проектування*

3. Патологічний ланцюжок розробки. Природа виникнення джерела цього ланцюжка полягає в процесі розробки, тобто дефектах розробки. Елементи такого ланцюжка будемо позначати індексом — буквами dev (development) (рис. 6). Множина елементів патологічного ланцюжка розробки, модифікації життєвого циклу дефекту ПЗ, матиме наступний вигляд (4):

$$SDLC_{dev} = \begin{Bmatrix} OM_{dev}, DM_{dev}, HD_{dev}, AD_{dev}, E_{dev}, \\ FR_{dev}, F_{dev}, CV_{dev}, AV_{dev}, UC_{dev}, UA_{dev} \end{Bmatrix}. \quad (4)$$

4. Патологічний ланцюжок взаємодії. Природа виникнення джерела цього ланцюжка полягає в процесі взаємодії з інформаційною системою, тобто в дефектах взаємодії. Елементи такого ланцюжка будемо позначати індексом — буквою і (interaction) (рис. 7). Множина елементів патологічного ланцюжка взаємодії, модифікації життєвого циклу дефекту ПЗ, матиме наступний вигляд (5):

$$SDLC_i = \{OM_i, DM_i, HD_i, AD_i, E_i, FR_i, F_i, CV_i, AV_i, UC_i, UA_i\}. \quad (5)$$

У загальному патологічному ланцюжку взаємодії можна виділити кілька його типів. До них відносяться:

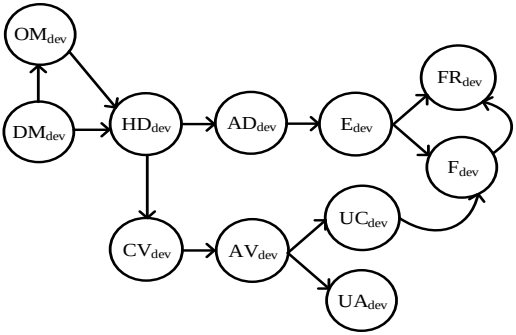

Рис. 6. *Патологічний ланцюжок розробки*

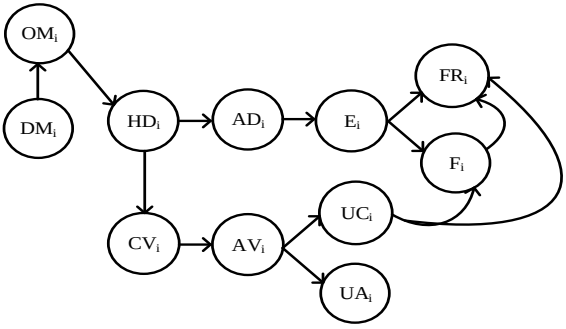

Рис. 7. *Патологічний ланцюжок взаємодії*

4.1. Патологічний ланцюжок взаємодії внаслідок фізичного впливу. Природа виникнення джерела цього ланцюжка в цілому аналогічна природі виникнення патологічного ланцюжка взаємодії, але з уточненням — внаслідок фізичного впливу. Елементи такого ланцюжка будемо позначати індексом — буквами pi (physical interaction) (рис. 8). Множина елементів патологічного ланцюжка взаємодії внаслідок фізичного впливу, модифікації життєвого циклу дефекту ПЗ, матиме наступний вигляд (6):

$$SDLC_{pi} = \begin{Bmatrix} OM_{pi}, DM_{pi}, HD_{pi}, AD_{pi}, E_{pi}, \\ FR_{pi}, F_{pi}, CV_{pi}, AV_{pi}, UC_{pi}, UA_{pi} \end{Bmatrix}. \quad (6)$$

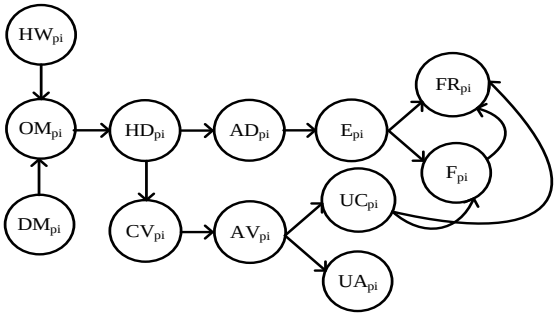

Рис. 8. *Патологічний ланцюжок взаємодії внаслідок фізичного впливу*

4.2. Патологічний ланцюжок взаємодії внаслідок інформаційного впливу. Природа виникнення джерела цього ланцюжка в цілому аналогічна природі виникнення патологічного ланцюжка взаємодії, але з уточненням — внаслідок інформаційного впливу. Елементи такого ланцюжка будемо позначати індексом — буквами ii (information interaction, ii) (рис. 9). Множина елементів патологічного ланцюжка взаємодії внаслідок інформаційного впливу, модифікації життєвого циклу дефекту ПЗ, матиме наступний вигляд (7):

$$SDLC_{ii} = \{OM_{ii}, DM_{ii}, HD_{ii}, AD_{ii}, E_{ii}, FR_{ii}, F_{ii}, CV_{ii}, AV_{ii}, UC_{ii}, UA_{ii}\}. \quad (7)$$

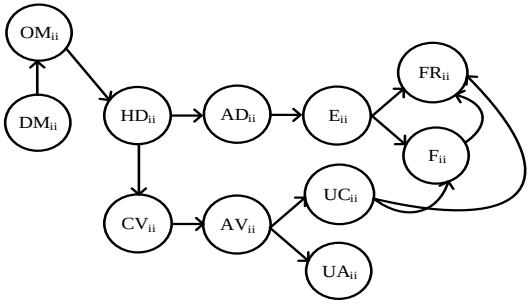

Рис. 9. *Патологічний ланцюжок взаємодії внаслідок інформаційного впливу*

Модель еволюції дефекту ПЗ з урахуванням V-подібної моделі розробки ПЗ. Варто відзначити, що ЖЦ дефекту ПЗ не завжди є лінійним (однорівневим), у тому вигляді, у якому він графічно представлений на рис. 1-9. Така нелінійність пов'язана з еволюцією дефектів ПЗ в рамках життєвого циклу розробки ПЗ. Наприклад, відповідно до V-подібної моделі дефект може з'явиться на першому етапі «1. Планування проекту та вимог» і бути дефектом вимоги ПЗ. При верифікації вимог на етапі «2. Аналіз вимог до продукту» такий дефект може бути невиявлений, тобто він стане прихованим дефектом і далі, еволюціонуючи, перейти на наступний етап розробки «3. Розробка архітектури» або ще далі — до етапу«5. Кодування». Така еволюція дефекту ПЗ може здійснюватися від етапу до етапу ЖЦ ПЗ, доти, доки він не проявиться при виконанні або тестуванні розроблюваного ПЗ (рис. 10). Таким чином дефект може еволюціонувати від HD_1 до HD_5. На етапі «5. Кодування» еволюція дефекту ПЗ припиняється і на наступних етапах «6. Модульне тестування», «7. Інтеграція і тестування»,«8. Системне приймальне тестування» і «9. Виробництво, експлуатація, супровід» він переходить від етапу до етапу тестування, як програмне забезпечення до моменту виявлення дефекту ПЗ.

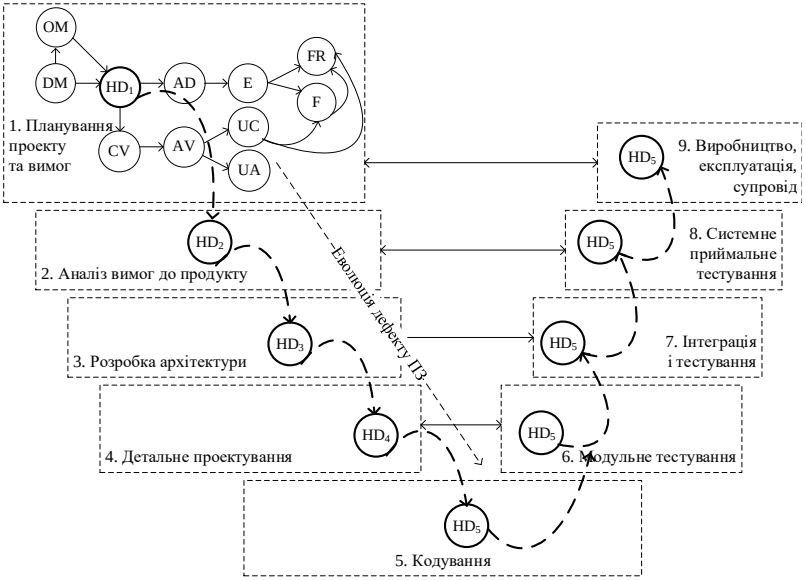

Рис. 10. *Графічне представлення еволюції дефекту ПЗ з урахуванням V-подібної моделі розробки ПЗ*

Висновки. Представлена в статті модель життєвого циклу програмного забезпечення дозволяє формально представити еволюцію

дефекту програмного забезпечення починаючи з причин, які привели до його виникнення і закінчуючи можливими наслідками. Більш детальне представлення моделі життєвого циклу дефекту ПЗ у вигляді множини патологічних ланцюжків дозволяє врахувати особливості різноманіття станів дефекту ПЗ.

Подальші дослідження доцільно спрямувати на розвиток моделі ЖЦ дефекту ПЗ в напрямку оцінювання якості програмного забезпечення на основі засіву дефектів ПЗ. У цьому напрямку окремим завданням є дослідження життєвого циклу дефектів при розробці та реалізації процедур засіву дефектів (Fault Injection Testing, FIT), які використовуються у Науково-виробничому підприємстві «Радій» для оцінки функціональної безпеки FPGA-проектів для локальних інформаційно-керуючих систем АЕС. При цьому сам ЖЦ дефекту ПЗ може уточнитися, а кількість його модифікацій збільшитися.

Список використаних джерел:

1. Software engineering. Report on a conference sponsored by the NATO science committee. Garmisch, Germany, 7-11 October, 1968. URL: http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF.
2. Wagner S. Software Product Quality Control: book. Heidelberg: Springer. 2013. 210 p.
3. Gao J., Zhang L., Zhao F., Zhai Ye. Research on Software Defect Classification. In proceedings of the IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC-2019). URL: https://ieeexplore.ieee.org/document/8729440/authors#authors.
4. TOP 5 Software Failures of 2018–2019. Website Checkio. URL: https://blog.checkio.org/%EF%B8%8F-top-5-software-failures-of-2018-2019-5-is-pretty-alarming-2a5400b01658.
5. Han W., Jiang H., Li W., Li Ye. A Summary of Software Defect Model. In proceedings of the IEEE 7th International Conference on Control and Automation (ICCA-2014). URL: https://ieeexplore.ieee.org/document/7026263.
6. Han W., Jiang H. Y., Lu T. B., Zhang X. Y., Li W. Software defect model based on similarity and association rule. International Journal of Multimedia and Ubiquitous Engineering. 2015. Vol. 10(7). P. 1-10. URL: http://gvpress.com/journals/IJMUE/vol10_no7/1.pdf.
7. Frattini F., Pietrantuono R., Russo S. Reproducibility of Software Bugs. In: Fiondella L., Puliafito A. (eds) Principles of Performance and Reliability Modeling and Evaluation. Springer Series in Reliability Engineering. Springer, Cham. 2016. P 551-565. URL: https://link.springer.com/chapter/-10.1007%2F978-3-319-30599-8_21.
8. Singh P. Learning from Software defect datasets. In proceedings of the 5th International Conference on Signal Processing, Computing and Control (ISPCC-2019). URL: https://ieeexplore.ieee.org/document/8988366.
9. Rahman A., Nurdatillah H. Defect Management Life Cycle Process for Software Quality Improvement. In proceedings of the 3rd International

Conference on Artificial Intelligence, Modelling & Simulation (AIMS-2015). URL: https://ieeexplore.ieee.org/document/7604582.
10. Koponen T. Life cycle of Defects in Open Source Software Projects. In: Damiani E., Fitzgerald B., Scacchi W., Scotto M., Succi G. (eds) Open Source Systems. OSS 2006. IFIP International Federation for Information Processing, vol 203. Springer, Boston, MA. URL: https://link.springer.com/-chapter/10.1007%2F0-387-34226-5_19.

LIFE CYCLE MODEL OF SOFTWARE DEFECT

The software development process includes a mandatory additional software quality assurance process, which is a set of measures covering all technological stages of software development, release and operation of information systems, carried out at different stages of the software life cycle, to ensure the required level of software quality. One of the main tasks of this process is to find and eliminate software defects. This work is devoted to a formal presentation of the life cycle of a software defect. The life cycle model of the software defect is considered as a chain that begins with a developer error and ends with a software failure. The article presents the general structure of the life cycle model of software defect, which includes developer error, operator error, hidden software defect, active software defect, calculation error, fault or failure, created vulnerability, activated vulnerability, unauthorized control and unauthorized access to data. This model is detailed in a set of pathological chains, which structurally represent modifications of the life cycle of software defect, taking into account the nature of the software defect itself. Among the pathological chains are the following: physical, design, development and interaction. The pathological chain of interaction is detailed in the pathological chain of interaction due to physical influence and the pathological chain of interaction due to informational influence. The already mentioned model of life cycle of software defect is projected on a V-shaped model of software development, which allows to present a kind of evolution of software defect at each stage of life cycle of software development separately and within the whole V-shaped model of software development.

Key words: *software quality, software defect, software defect life cycle, software life cycle, pathological chains.*

Отримано: 19.09.2020