Технічні науки

УДК 004.41:519.85

Григорчук Галина Василівна
доктор філософії, доцент,
доцент кафедри прикладного програмування та обчислення
Івано-Франківський національний технічний університет нафти і газу
Grygorchuk Galina
PhD, Associated Professor,
Associated Professor of the Department of Applied Programming and Computing
Ivano-Frankivsk National Technical University of Oil and Gas

Романюк Богдан Андрійович
здобувач освіти
Івано-Франківського національного технічного університету нафти і газу
Romaniuk Bohdan
Student of the
Ivano-Frankivsk National Technical University of Oil and Gas

Царева Олександра Степанівна
асистент кафедри прикладного програмування та обчислення
Івано-Франківський національний технічний університет нафти і газу
Tsareva Oleksandra
Assistant of the Department of Applied Programming and Computing
Ivano-Frankivsk National Technical University of Oil and Gas

ЗАСТОСУВАННЯ КОНЦЕПЦІЇ ХАОСУ В ПРОГРАМУВАННІ, МАТЕМАТИЦІ ТА КОМАНДНИХ СЕРЕДОВИЩАХ
APPLYING THE CONCEPT OF CHAOS IN PROGRAMMING, MATHEMATICS, AND TEAM ENVIRONMENTS

Анотація. *Ця стаття досліджує використання концепції хаосу в програмуванні та командних середовищах. Вона розглядає систему хаосу, яка охоплює невизначеність і складність, підкреслюючи гнучкість і адаптивність у розробці програмного забезпечення. Особлива увага приділяється схемі хаосу та його застосуванню в математиці, включаючи відомий дивний атрактор Лоренца, який є прикладом хаотичної поведінки в системі диференціальних рівнянь. Стаття аналізує, як хаос впливає на життєвий цикл програмного забезпечення. Стаття також надає практичні поради щодо впровадження стратегії хаосу. Серед них – встановлення чітких каналів зв'язку, що сприяє ефективній комунікації між командами, а також створення культури експериментування, що спонукає до інноваційного мислення та використання нових підходів у розробці програмного забезпечення. Особлива увага приділяється психологічній безпеці, яка стимулює команди до спроби нових ідей та ризикування в пошуку кращих рішень.*

Ключові слова: *хаос, система хаосу, атрактор Лоренца, модель хаосу, схема хаосу, стратегія хаосу, життєвий цикл ПЗ, командні середовища, хаос в диференціальних рівняннях.*

Summary. *This article explores the use of the concept of chaos in programming and team environments. It considers a system of chaos that embraces uncertainty and complexity, emphasizing flexibility and adaptability in software development. Special attention is paid to the scheme of chaos and its applications in mathematics, including the famous strange Lorentz attractor, which is an example of chaotic behavior in a system of differential equations. The article analyzes how chaos affects the software life cycle. The article also provides practical advice on implementing a chaos strategy. Among them are the establishment of clear communication channels, which promotes effective communication between teams, as well as the creation of a culture of*

experimentation, which encourages innovative thinking and the use of new approaches in software development. Special attention is paid to psychological safety, which encourages teams to try new ideas and take risks in search of better solutions.

Key words: *chaos, chaos system, Lorentz attractor, chaos model, chaos scheme, chaos strategy, software life cycle, command environments, chaos in differential equations.*

Вступ. У сучасному складному технологічному середовищі, що швидко розвивається, актуальність моделі хаосу стає все більш очевидною. Оскільки традиційним лінійним підходам важко йти в ногу з вимогами, що постійно змінюються, і непередбачуваними проблемами, модель хаосу пропонує новий погляд. Охоплюючи невизначеність і складність, команди можуть використовувати потенціал моделі хаосу для навігації в складній взаємодії динамічних систем [1; 2].

Основна увага цього дослідження зосереджена на моделі хаосу та її застосуванні в математиці та програмуванні, зокрема в командному середовищі. Модель хаосу, заснована на теорії складних систем, охоплює невизначеність і складність, кидаючи виклик традиційним лінійним підходам. У царстві математики прикладом хаосу є такі явища, як дивний атрактор Лоренца [3], який демонструє складні закономірності, які можуть виникати внаслідок, здавалося б, випадкової поведінки. Стаття заглиблюється в графічне представлення моделей хаосу, фіксуючи їх динамічну та візуально захоплюючу природу. У ньому також обговорюються переваги та недоліки прийняття моделі хаосу, підкреслюється її потенціал для сприяння творчості, навичкам вирішення проблем і співпраці, визнаючи при цьому проблеми управління невизначеністю та підтримки стабільності. Крім того, у статті розглядається, як модель хаосу можна застосувати до життєвого циклу

розробки програмного забезпечення, пропонуючи розуміння її ітеративної та адаптивної природи. Вивчаючи та використовуючи модель хаосу, команди можуть долати складні виклики та використовувати її потенціал для стимулювання інновацій та стійкості перед обличчям невизначеності.

Мета роботи. Виявити потенціал та переваги використання моделі хаосу в програмуванні та командних середовищах, зокрема в командному середовищі розробки програмного забезпечення, а також дослідити застосування моделі хаосу в математиці, зокрема вивчення дивного атрактора Лоренца. Проаналізувати стратегії хаосу та їх вплив на креативність, навички вирішення проблем та співпрацю в командному середовищі. Дослідити графічне представлення моделі хаосу та надати практичні поради щодо впровадження стратегії хаосу, зокрема встановлення чітких каналів зв'язку та сприяння культурі експериментування. Розглянути модель хаосу в контексті життєвого циклу розробки програмного забезпечення та визначити, як вона сприяє інноваціям та стійкості в постійно змінюваних умовах.

Поняття хаосу

У контексті повсякденної мови «хаос» зазвичай означає повний безлад. Проте в теорії хаосу термін «хаотичний» визначається більш конкретно. Хоча загальноприйнятого математичного визначення хаосу не існує, він зазвичай характеризується певними властивостями, якими має володіти динамічна система, щоб класифікувати її як хаотичну.

Можна навести такі властивості, завдяки яким характеризується система хаосу в математиці:

- Вона має бути чутливою до початкових умов.
- Вона повинна мати властивість топологічного змішування.
- Її періодичні орбіти мають бути скрізь щільними.

Точніші математичні умови виникнення хаосу виглядають так [2].

Система повинна мати нелінійні характеристики, бути глобально стійкою, але мати хоча б одну нестійку точку рівноваги коливального типу, при цьому фрактальна розмірність системи повинна бути не менше ніж 1,5.

Яскравим прикладом використанні хаосу в математиці є дивний атрактор Лоренца. Для початку розглянемо рівняння Дуффинга для вимушених коливань (1):

$$mx'' + cx' + kx + \beta x^3 = F_0 cos\omega t \qquad (1)$$

Підстановка рівняння Дуффинга для вимушених коливань (1) призводять до двовимірної нелінійної системи диференціальних рівнянь. Тривимірну нелінійну систему диференціальних рівнянь стосовно завдань метеорології розглядав Е.Н. Лоренц (2):

$$\begin{aligned} \frac{dx}{dt} &= -sx + sy, \\ \frac{dy}{dt} &= -x + rx - y, \\ \frac{dz}{dt} &= xy - dz \end{aligned} \qquad (2)$$

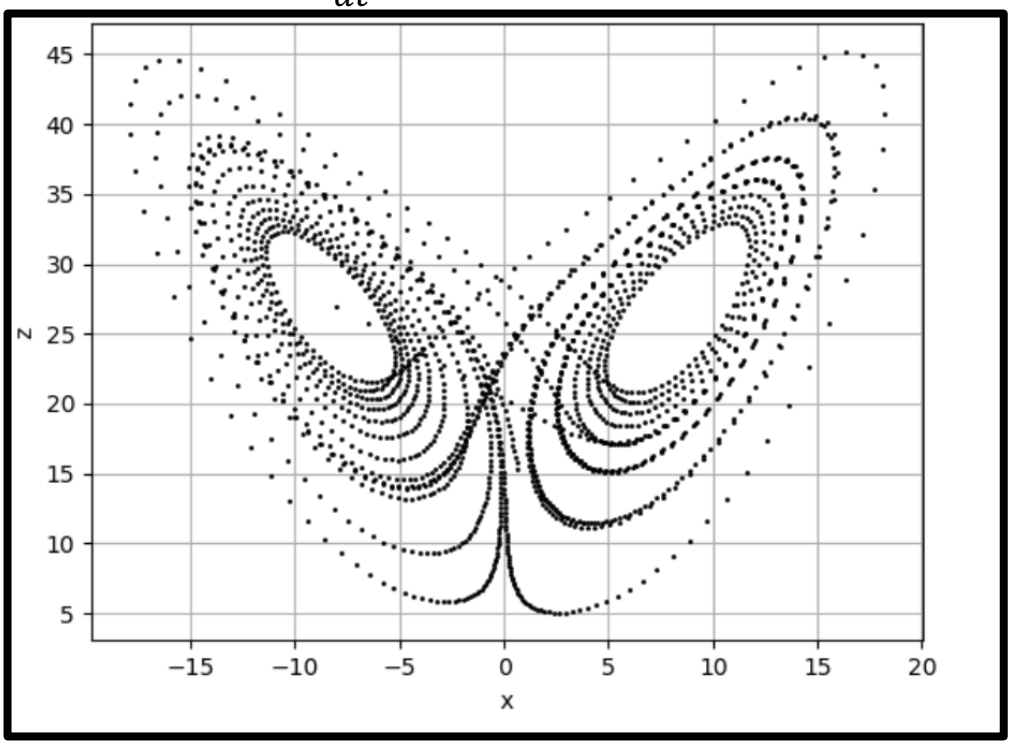

Рис. 1. Проєкція траєкторії Лоренца на площину *xz*

Рішення системи (2) краще розглядати у проекції на одну із трьох площин. Вже було написано програму чисельного інтегрування при значеннях параметрів $b = \frac{8}{3}$, $s = 10$, $r = 28$ та початкових умовах $x(0) = -8$, $y(0) = 8$, $z(0) = 27$.

В результаті виконання програми отримуємо проєкцію траєкторії Лоренца на площині xz (рис. 1).

Розглядаючи зображення на графіці у часі, можна припустити, що точка P(x(t), y(t), z(t)) здійснює випадкове число коливань то справа, то зліва. Для метеорологічного застосування системи Лоренца, після випадкової кількості ясних днів, слідує випадкове число дощових днів.

Поняття моделі хаосу

У сфері обчислень модель хаосу служить підходом до розробки програмного забезпечення. Л. Б. С. Ракун [1; 4], творець цієї моделі, підкреслює, що звичайні моделі управління проєктами, такі як спіральна модель і каскадна модель, можуть полегшити графік і управління персоналом, але їм часто бракує ефективних стратегій для усунення помилок, вирішення технічних проблем, дотримання кінцевих термінів і реагування на вимоги клієнтів.

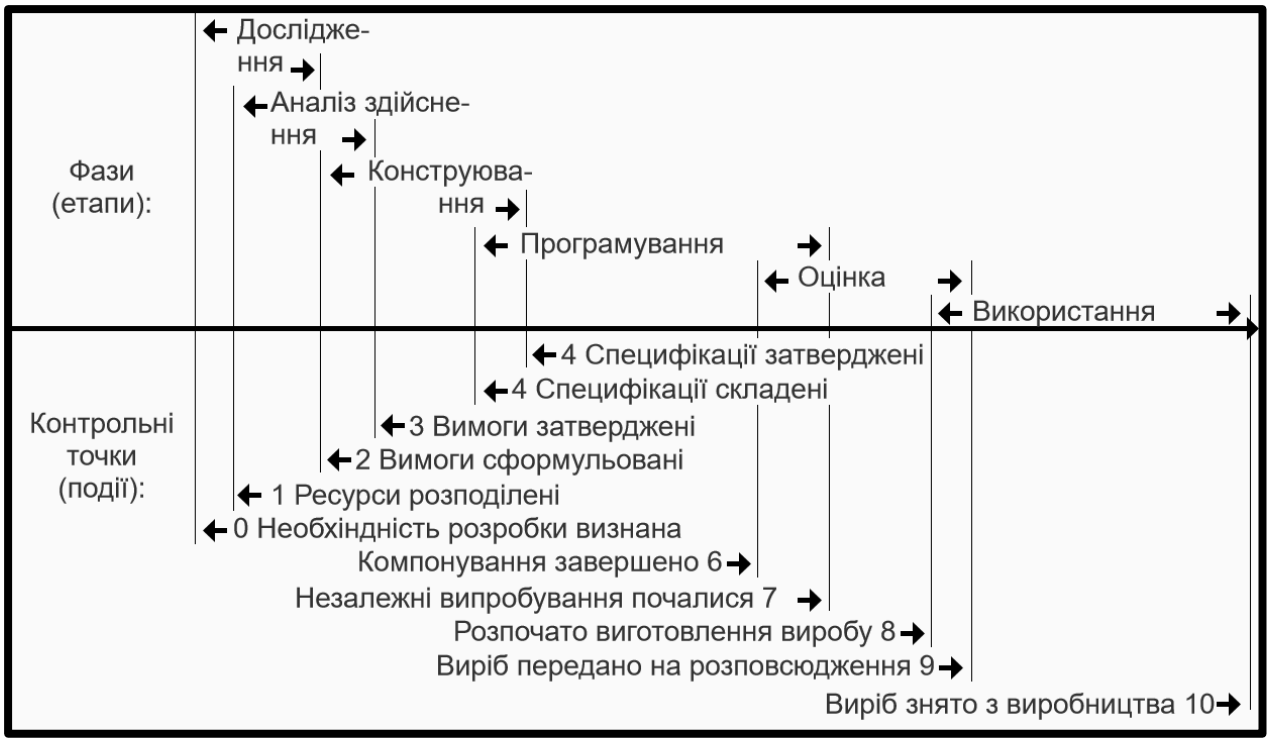

Рис. 2. Схема хаосу

Модель хаосу має на меті подолати ці прогалини та надати розуміння цих обмежень.

Було створено схему хаосу для розподілення етапів розробки ПЗ і зображено її у графічному вигляді (рис. 2):

На цьому рис. 2 можна побачити модель хаосу зображену в графічному вигляді [5]. Велика чорна лінія являє собою лінію часу повної реалізації продукту, враховуючи кожний етап та перетинаючи при цьому всі контрольні точки.

Для прикладу візьмемо перший етап «Дослідження». Вертикальні лінії показують нам межі даного етапу і нижче відповідно контрольні точки, які під час певного етапу перетинаються. У випадку фази «Дослідження» потрібні здійснитися події «Визначити необхідність розробки» та «Розподілити ресурси». Інші етапи відбуваються за схожим сценарієм.

Переваги та недоліки моделі хаосу

До переваг моделі хаосу можна віднести те, що:

- Враховується взаємодія між членами команди при внесенні змін в коді;
- Обмежується ризик надмірного проєктування рішення;
- Прозорість між бажаннями керівництва високого рівня і розумінням командою розробників проблем і пріоритетів.

Недоліком цієї системи є:

- Критична необхідність створити єдиний дизайн на рівні коду, який необхідно виконати для задоволення потреб на рівні програми.

Життєвий цикл програмного забезпечення

Життєвий цикл програмного забезпечення – це період часу, який починається з моменту прийняття рішення про створення програмного продукту і закінчується в момент його повного вилучення та експлуатації.

Модель хаосу зазначає, що фази життєвого циклу [6] поширюються на всі рівні проєкту, від усього проекту загалом, до окремого рядка коду. Проєкт загалом, системи, модулі, функції та рядки коду мають бути визначені, реалізовані та інтегровані.

Одна важлива зміна в перспективі — це чи можуть проєкти бути представлені як цілісні модулі або повинні бути представлені частинами. Мається на увазі, що розробка проводиться частинами, модулями. Розробляється деякий малий блок та перевіряється коректність його роботи. Після розробки необхідних блоків/модулів вони інтегруються у складнішу і більшу систему. Поведінка складної системи виходить із комбінованої поведінки складових її менших блоків.

Стратегія хаосу

Стратегія хаосу — це стратегія розробки програмного забезпечення на основі моделі хаосу. Головне правило - це завжди вирішувати найважливіше питання першим.

- *Проблема* - це незавершене завдання програмування.

- *Найважливіша проблема* - це комбінація великого розміру, терміновості та стійкості.
- *Завдання великого розміру* цінні для користувачів настільки, наскільки вони функціональні.
- *Термінові завдання* своєчасні настільки, наскільки має бути, інакше затримується решта роботи.
- *Стійкі завдання* перевірені та випробувані. Розробники можуть сфокусуватися на іншому.
- *Вирішити* означає привести все до стану стабільності.

Стратегія хаосу схожа на шлях, за яким програмісти працюють в самому кінці проекту, коли у них є список помилок для виправлення та можливість для творчості. Зазвичай, хтось розставляє пріоритет приватним завданням, що залишилися, і програмісти усувають їх по одному. Стратегія хаосу стверджує, що це єдиний коректний шлях виконання роботи.

Зв'язок із теорією хаосу

Можна врахувати декілька зв'язків із теорією Хаосу. До них відносяться:

- Модель хаосу може допомогти пояснити, чому програмне забезпечення має тенденцію бути настільки непередбачуваним.
- Вона пояснює чому такі високорівневі концепції, як архітектура ЕОМ неспроможні розглядатися незалежно від низькорівневих рядків коду.
- Вона забезпечує хитрощами для пояснення, що робити наступним, в умовах стратегії хаосу.

Висновки. Під час дослідження цієї теми стало очевидно, що модель хаосу з її застосуванням у математиці, програмуванні та налаштуваннях команди пропонує цінну інформацію про навігацію в складних і непередбачуваних середовищах. Дослідження теорії хаосу, прикладом якого є такі явища, як дивний атрактор Лоренца, забезпечує глибше

розуміння складних закономірностей, які можуть виникати з, здавалося б, хаотичних систем.

Графічне представлення моделей хаосу відображає динамічну природу складних систем, роблячи їх візуально захоплюючими інструментами для аналізу та дослідження. Однак важливо враховувати як плюси, так і мінуси прийняття моделі хаосу. З одного боку, це сприяє розвитку креативності, навичок вирішення проблем і співпраці всередині команд, створюючи інноваційні рішення та адаптивність. З іншого боку, управління невизначеністю та підтримання стабільності може створити проблеми, які потребують ретельного розгляду.

У контексті розробки програмного забезпечення застосування моделі хаосу до життєвого циклу програмного забезпечення вводить ітераційний та адаптивний підхід. Це заохочує безперервне навчання, експериментування та здатність ефективно реагувати на мінливі вимоги та нові ідеї.

Підсумовуючи, модель хаосу пропонує цінну структуру для розуміння та навігації складних систем у математиці, програмуванні та налаштуваннях команди. Прийнявши невизначеність і складність, команди можуть використовувати силу хаосу для стимулювання інновацій, стійкості та успішних результатів. Проте вкрай важливо збалансувати переваги та проблеми, пов'язані з ефективним управлінням хаосом. Загалом, модель хаосу забезпечує переконливий підхід до вирішення складнощів нашого постійно мінливого світу.

Література

1. Raccoon. The Chaos Model and the Chaos Life Cycle, in ACM Software Engineering Notes. January 1995. Vol. 20, № 1. P. 55-66, ACM Press.
2. Holmes T. What is the Chaos Model? Easy Tech Junkie. URL: https://www.easytechjunkie.com/what-is-the-chaos-model.htm (дата звернення: 20.03.2023)
3. Рудик О. Система Лоренца як приклад динамічної системи зі складною поведінкою. URL: http://www.kievoi.ippo.kubg.edu.ua/kievoi/dynsys/lorence.html (дата звернення: 20.03.2023)
4. Pressman R. Software Engineering: A Practitioner's Approach 4th edition. 1997. P. 29–30, McGraw Hill.
5. Chaos model. Computer Hope. 2022. URL: https://www.computerhope.com/jargon/c/chaos-model.htm#:~:text=The%20chaos%20model%20is%20an,forming%20an%20overall%20superior%20strategy (дата звернення: 23.03.2023)
6. Популярні життєві цикли розробки ПЗ. Тренінговий центр QATestLab. 2023. URL: https://training.qatestlab.com/blog/technical-articles/popular-software-development-life-cycles/ (дата звернення: 24.03.2023)