

Завдяки такому підходу результат проекту повністю відповідає потребам та очікуваним результатам. Scrum дозволяє якісно організувати роботу команди, впливає на продуктивність, та враховує інтереси усіх членів команди.

В рамках дослідження цей підхід можна цілком адаптувати в освітній процес, що допоможе досягнути нових цілей як учнями, так і педагогами.

Список використаних джерел

1. Дика Н., Глазова О. Організація проектної діяльності учнів закладів загальної середньої освіти у процесі вивчення неологізмів. *Педагогічний процес: теорія і практика. Серія: Педагогіка*. 2019. Вип. № 3–4(66–67). С. 145–152.
2. Проектне навчання: коротко про головне. *НУШ – нова українська школа*. URL: <https://nus.org.ua/view/proektne-navchannya-korotko-pro-golovne> (дата звернення: 04.04.2024).
3. Скасків Г. М. Методичні аспекти використання проектних технологій у системі дистанційного навчання у процесі вивчення інформатики. *Наук. зап. Терноп. нац. пед. ун-ту ім. В. Гнатюка. Сер. Педагогіка*. 2011. Вип. 1. С. 108–114.
4. Теоретичні й методичні основи компетентнісного підходу як засобу досягнення освітніх завдань початкової школи. *Нова українська школа*. URL: <https://nuschool.eu/lessons/world/1-2klas/2.html> (дата звернення: 04.04.2024).
5. Erdogan N., Bozeman T.D. Models of Project-based Learning for the 21st Century. *A Practice-based Model of STEM Teaching*. 2015. P. 31–42.

АКТУАЛЬНІСТЬ ВИКЛАДАННЯ ЛОГІК НЕКОРЕКТНОСТІ В ЗАКЛАДАХ ВИЩОЇ ОСВІТИ

Криволап Андрій Володимирович

кандидат фізико-математичних наук, асистент кафедри теорії та технології програмування,
Київський національний університет імені Тараса Шевченка,
andriikryvolap@knu.ua

Русіна Наталія Геннадіївна

кандидат педагогічних наук, доцент кафедри теорії та технології програмування,
Київський національний університет імені Тараса Шевченка,
rusina@knu.ua

Згідно сучасних підходів до розробки програмних продуктів, в розрізі забезпечення коректності та якості, все більшої ролі набувають засоби статичного аналізу коду, та пошуку потенційних помилок в коді. Разом з тим, задача верифікації, все ще потребує значних ресурсів, що часто перевищують ресурси, потрібні на розробку власне системи, що верифікується. Логіки некоректності [4] є теоретичною основою систем направлених на пошук потенційних помилок та систем аналізу вразливостей програмних систем. Саме тому виникає питання доцільності їх розгляду в рамках навчальних дисциплін з формальних методів, верифікації програмних систем, та кібербезпеки, що є частиною підготовки майбутніх фахівців ІТ-галузі.

Коли мова йде про формальну верифікацію, одним з найбільш часто застосовуваних інструментів є логіка Флойда-Хоара [1] та її подальші розширення. Це зумовлено використанням трійок Хоара, в яких специфікація програми задається за допомогою передумови, вираженої певною формулою, тексту програми та післяумови, що також описується формулою. В своїй роботі

Тоні Хоар [1] навів визначення коректності такої трійки та запропонував відповідну систему виводу для простої імперативної мови програмування з циклом та умовним оператором. Це дозволило формально доводити потрібні властивості програм, та згодом були запропоновані різноманітні розширення логіки Флойда-Хоара, разом з відповідними системами виводу, для роботи з локальними та глобальними змінними, вказівниками, паралельними процесами.

Не зважаючи на тривалу історію досліджень та велику кількість розширень, деякі питання, що стримують впровадження та широке застосування верифікації до програмних систем залишаються невирішеними. Однією з таких проблем є необхідність великої кількості додаткових анотацій, що описували б проміжні умови при доведенні коректності, зважаючи на те, що їх автоматична генерація не є можливою. Лише інтерактивні напівавтоматичні засоби існують. Іншою проблемою, що впливає з визначення коректності трійки Хоара, є схильність до знаходження значної кількості хибно-позитивних помилок. Трійка є коректною, якщо програма, починаючи свою роботу в стані, для якого істинна передумова, або зациклюється, або завершує роботу в стані, в якому істинна післяумова. Таке визначення орієнтовано саме на проблему верифікацій, адже не допускає загалом помилок. В такому разі не можливо уникнути хибно-позитивних помилок. Це можливо, бо для станів, в яких істинна післяумова, не гарантується, що такий стан взагалі досяжний. Звідки впливає, що логіка Флойда-Хоара, хоч і пристосована для верифікації, не є оптимальним рішенням для процесу пошуку помилок, або вразливостей системи. Побудовані на ній системи гарантують відсутність помилок, проте не гарантують, що визначені помилкові стани, дійсно такими є, а не недосяжними станами, отриманими лише за рахунок природи правил системи виводу для логіки, що є схильними до над-апроксимації.

Відносна складність верифікації програмних систем та відповідних засобів, що часто потребують більшої кількості рядків опису умов, ніж кількість рядків коду системи, що верифікується, призвела до переорієнтації зусиль по забезпеченню коректності та якості програмного забезпечення на використання засобів пошуку помилок, радше ніж доведення їх відсутності. Масштаб впровадження таких систем лише зростає, яскравим прикладом є система Infer від Facebook [2]. Дана система заснована на логіці розділення, що є розширенням логіки Флойда-Хоара, проте дозволяє більш локальні твердження та їх доведення окремо від глобального стану програми. Остання властивість дуже корелює з сучасними тенденціями розробки програмного забезпечення з використанням гнучких методологій та принципами постійного впровадження за рахунок невеликих змін. Саме під час розробки Infer, Пітером О'Хеарном було запропоновано визначення логіки некоректності [4], відповідних специфікацій, та їх істинності. При цьому, найбільшу увагу приділено саме досяжності станів післяумови, адже окрім дослідження факту наявності помилки, важливим є також відтворення дій, що призводять до помилки. Такий підхід, зокрема довів свою актуальність при дослідженні питання вразливості системи, що є ключовим в сфері кібербезпеки. В подальшому ідеї логіки розділення та логіки некоректності були поєднані [3], та на їх основі продовжується розвиток системи Infer,

поєднавши можливості дослідження локальних властивостей з пошуком сценаріїв, що призводять до помилок.

Зменшення кількості помилок та вразливостей завжди було одним з пріоритетних напрямів розвитку комп'ютерної інженерії. В даній сфері важливою є наявність засобів, що є теоретично обґрунтованими, та мають відповідну наукову основу. Висока ефективність та масштабованість систем статичного аналізу коду та інших засобів визначення помилок, зумовлює актуальність подальшого їх розвитку, адже вони також орієнтовані на знаходження вразливостей, та потенційних сценаріїв, що до них призводять. Тому потрібні інженери з відповідною кваліфікацією, та подальші наукові розробки. Так як логіка некоректності є основою таких рішень і є дуальною до логіки Флойда-Хоара, що широко вивчається в межах навчальних дисциплін ІТ-галузі, доцільним є включення її розгляду до програм дисциплін верифікації програмних систем, кібербезпеки, формальних методів, та подальших досліджень.

Список використаних джерел

1. Apt K. Fifty years of Hoare's logic. K. Apt, E. Olderog. *Formal Aspects Comput*, 2019. № 6. P. 751–807.
2. Facebook. Infer. URL: <https://fbinfer.com/docs/separation-logic-and-bi-abduction>.
3. Local Reasoning About the Presence of Bugs: Incorrectness Separation Logic. [A. Raad, J. Berdine, H. Dang та ін.]. *Lecture Notes in Computer Science*, 2020. № 12225. С. 225–252.
4. O'Hearn P. Incorrectness logic. Peter W. O'Hearn. *Proc. ACM Program. Lang.* 2020. № 4. С. 1–32.

ФОРМУВАННЯ У МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ ГОТОВНОСТІ ДО ВИКОРИСТАННЯ ПЛАТФОРМ РОЗРОБКИ Й ОЦІНКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кубік Михайло Анатолійович

здобувач третього (освітньо-наукового) рівня вищої освіти спеціальності
011 Освітні, педагогічні науки,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
kub@tnpu.edu.ua

Мартинюк Сергій Володимирович

кандидат фізико-математичних наук, доцент кафедри інформатики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
sergmart65@tnpu.edu.ua

Сучасний світ переживає стрімкий розвиток технологій, що вимагає постійного оновлення й адаптації освітньої системи до нових вимог. У цьому контексті особливо актуальною є підготовка майбутніх учителів інформатики, які мають бути готові до використання платформ розробки й оцінки програмного забезпечення. Тому важливо оцінити важливість такої підготовки та методи формування готовності майбутніх учителів до цих навичок.

В умовах інформатизації суспільства інформатика як наукова дисципліна стає однією з фундаментальних галузей наукового знання, у рамках якої вивчаються інформаційні процеси, методи та засоби отримання, опрацювання,