УДК 004.05

В. М. Бажан[1]
О. В. Романюк[2]

ВІДСТЕЖЕННЯ ДЕФЕКТІВ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

[1] ТОВ ВКФ «СЕНС ЛТД»,
[2] Вінницький національний технічний університет

Анотація
Стаття описує основні інструменти та методи відстеження дефектів, а також наводить приклади випадків, коли відсутність відстеження дефектів призводила до серйозних наслідків для користувачів та розробників.
Ключові слова: розробка програмного забезпечення, дефекти, відстеження дефектів, якість програмного забезпечення, інструменти відстеження дефектів.

Abstract
The article describes the main tools and methods of defect tracking, and also provides examples of cases where the absence of defect tracking led to serious consequences for users and developers.
Keywords: software development, defects, defect tracking, software quality, defect tracking tools.

Вступ

Розробка програмного забезпечення є складним та багатоетапним процесом, під час якого можуть виникати різноманітні дефекти. Ці дефекти можуть мати серйозний вплив на роботу програми, та навіть призвести до непередбачуваних наслідків для користувачів. Тому відстеження дефектів є надзвичайно важливою складовою процесу розробки програмного забезпечення, що відбувається в рамках етапу тестування, на який за широко відомою оцінкою трудомісткості між фазами створення програмного забезпечення припадає не менше 40% зусиль [1]. І це зрозуміло, адже за підрахунками на етапі тестування сучасних комерційних програм необхідно виконувати в середньому від 2 до 10 тисяч тест-кейсів [2] і при цьому в програмах з більш, ніж 1 млн. рядків коду, може бути знайдено до 20 тис. помилок [3].

Тому питання відстеження дефектів в процесі розробки програмного забезпечення, а саме основні інструменти та методи відстеження дефектів та наслідки через відсутність відстеження дефектів є досить актуальними.

Результати дослідження

Основними інструментами та методами відстеження дефектів в процесі розробки програмного забезпечення є:

1. Системи керування помилками (Bug Tracking Systems) – це програмні засоби, які дозволяють стежити за дефектами та їх статусом, присвоювати пріоритети та відслідковувати процес їх виправлення. Найпоширеніші системи керування помилками – Jira, Bugzilla, Redmine, Trac, MantisBT [3, 4].
2. Тестування програмного забезпечення включає проведення тестів, які дозволяють виявляти дефекти в програмному забезпеченні та перевіряти, чи виконуються вимоги до функціональності та якості продукту. Тестування може бути проведене як вручну, так і автоматизовано [5].
3. Контроль якості коду (Code Quality Control) включає в себе проведення аналізу коду з метою виявлення потенційних проблем, таких як витік пам'яті, неправильне використання пам'яті, дублювання коду, неефективність та інші.
4. Аналіз використання програми (Application Usage Analysis) – це метод відстеження дефектів, який включає аналіз даних про використання програми користувачами з метою виявлення

проблем, які можуть виникати в реальному середовищі використання.

5. Code Review – це метод, в якому члени команди розглядають код, що написаний іншим членом команди з метою виявлення потенційних проблем та дефектів [5].
6. Continuous Integration – це метод, в якому код змінюється та тестується автоматично при кожній зміні. Це дозволяє швидко виявляти та виправляти дефекти, які можуть виникати при зміні коду.

Використання цих інструментів та методів дозволяє ефективно відстежувати та усувати дефекти на ранніх етапах розробки, що дозволяє забезпечити високу якість продукту та зменшити ризики для користувачів та розробки.

Відсутність відстеження дефектів може призводити до серйозних наслідків для користувачів та розробників, оскільки це може призвести до поширення дефектів у продукті та зниження його якості. Наведемо кілька прикладів:

- Відсутність відстеження дефектів в медичному ПЗ може призвести до помилок у діагностиці та лікуванні хворих, що може мати серйозні наслідки для їхнього здоров'я та життя.
- Відсутність відстеження дефектів у фінансових системах може призвести до втрат коштів або порушення законодавства.
- Відсутність відстеження дефектів у вбудованих системах у автомобілях може призвести до аварій та серйозних наслідків для життя та здоров'я водіїв та пасажирів.
- Відсутність відстеження дефектів у веб-додатках може призвести до небезпечних вразливостей та атак хакерів.
- Відсутність відстеження дефектів у вбудованих системах в літаках може призвести до аварій та катастроф, що може мати серйозні наслідки для життя та здоров'я пасажирів та екіпажу.

Одним з найбільш відомих прикладів серйозних наслідків від відсутності відстеження дефектів є катастрофа літака Boeing 737 MAX в жовтні 2018 року. Відомо, що в цій катастрофі загинули 189 осіб через проблему з автоматичною системою керування, яка не враховувала дані з нового датчика. Дана проблема пов'язана зі збоєм в системі автоматичного підтримання вертикальної швидкості, яка призвела до аварії літака. Дослідження показали, що ця проблема могла бути виявлена та виправлена під час випробувань, якби було відстеження дефектів та відповідний аналіз. Інший приклад стосується соціальної мережі Facebook, яка стала жертвою атаки хакерів у 2018 році. У результаті цієї атаки зловмисники отримали доступ до особистих даних майже 30 мільйонів користувачів. Пізніше виявилося, що дана атака була можлива через вразливість в програмному забезпеченні, яку не виявили та не виправили вчасно через відсутність відстеження дефектів. Інший приклад стосується злому транспортної системи міста Сан-Франциско в 2018 році, яка була об'єктом кібератаки. Хакери змогли отримати доступ до системи та вимагали виплату викупу. Ця атака була можлива через вразливість у програмному забезпеченні, яку не було виявлено через відсутність відстеження дефектів. Ці приклади показують, як відсутність відстеження дефектів може призводити до серйозних наслідків для користувачів та розробників.

Висновки

Встановлено, що відстеження дефектів є важливою складовою процесу розробки програмного забезпечення, оскільки це дозволяє розробникам виявляти та виправляти помилки на ранніх етапах розробки та перед випуском продукту на ринок. З прикладів, наведених у статті, можна побачити, що відсутність відстеження дефектів може призводити до серйозних проблем, таких як витрати на виправлення помилок, втрату довіри користувачів та навіть загрозу безпеці. Тому, відстеження дефектів має бути не тільки важливою частиною процесу розробки програмного забезпечення, але й необхідним елементом для забезпечення безпеки та якості продукту.

Отже, розуміння важливості відстеження дефектів та використання відповідних інструментів та методів може допомогти забезпечити успішну розробку та випуск високоякісного програмного забезпечення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Поморова О. В. Сучасні проблеми оцінювання якості програмного забезпечення / О. В. Поморова, Т. О. Говорущенко // Радіоелектронні і комп’ютерні системи. – 2013. – № 5. – С. 319–327.

2. Романюк О. В.,Латуша А. В. Аналіз основних проблем автоматизованого тестування програмного забезпечення // Електронні інформаційні ресурси: створення, використання, доступ. Пам'яті Олексія Петровича Стахова. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 9-10 листопада 2021 р. – Суми/Вінниця: НІКО/ВНТУ, 2021. – 224 с. – ISBN 978-617-7422-16-6. – С. 174-176.

3. Побігай О. О. Порівняльний аналіз систем відстеження дефектів програмного забезпечення [Електронний ресурс] / О. О. Побігай // Матеріали XLV Науково-технічної конференції ВНТУ, Вінниця, 23-24 березня 2016 р. - Електрон. текст. дані. - 2016. - Режим доступу : http://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2016/paper/view/1088.

4. Gerdisch E. 20 Best Bug Tracking Software, Tools, & Solutions for 2023 [Електронний ресурс] / Evan Gerdisch – Режим доступу до ресурсу: https://clickup.com/blog/bug-tracking-software/.

5. Василевський, В.О, Романюк, О.В. "НЕДОЛІКИ ВИКОРИСТАННЯ МОДУЛЬНОГО ТЕСТУВАННЯ ЯК ОСНОВНОЇ ТЕХНОЛОГІЇ ТЕСТУВАННЯ" / Тези доповідей XI Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології – 2020 (ІКТ-2020)», м. Житомир, 09 - 11 квітня 2020 р. – Житомир: Житомирська політехніка, 2020. – С.14-15.

Бажан Вікторія Михайлівна — студентка групи 2ПІ-19б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, email: bazhan.viktoriya@icloud.com

Романюк Оксана Володимирівна — доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця,email: oroman@vntu.edu.ua

Bazhan Victoria M. — Department Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email:bazhan.viktoriya@icloud.com

Romanyuk Oksana V. — Associate Profesor of Software, Vinnytsia National Technical University, Vinnytsia, email:oroman@vntu.edu.ua.