

УДК 004.891.3: 004.3

НЕЙРОМЕРЕЖНИЙ МЕТОД ОЦІНЮВАННЯ ТА ПРОГНОЗУВАННЯ ЯКОСТІ СИСТЕМНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Т.О.ГОВОРУЩЕНКО, В.В.КОЗАЧУК, В.І.ГРИБИНЧУК

Хмельницький національний університет

У даній статті автори проаналізували сучасний стан галузі та метричну інформацію для оцінювання якості системного програмного забезпечення (СПЗ), а також запропонували нейромережний метод оцінювання результатів проектування і прогнозування складності та якості СПЗ.

In this article, the authors analyzed the current situation of the software quality assessment field and the metric for system software quality evaluating and proposed the neural network method for estimation of design results and prognostication of system software complexity and quality.

Ключові слова: програмне забезпечення, системне програмне забезпечення, метрики складності, метрики якості

Вступ. Програмне забезпечення (ПЗ) поділяється на дві великі групи: системне ПЗ та прикладне ПЗ. Системне програмне забезпечення (СПЗ) – це комплекс програм, які забезпечують керування компонентами комп'ютерної системи (процесором, оперативною пам'яттю, пристроями введення-виведення, мережевим обладнанням), виступаючи своєрідним прошарком (інтерфейсом), з одного боку якого – апаратура, а з іншого – додатки користувача [1]. СПЗ не вирішує конкретних практичних задач, а лише забезпечує роботу інших програм, надаючи їм сервісні функції, які абстрагують деталі апаратної та мікропрограмної реалізації обчислювальної системи, а також керує апаратними ресурсами обчислювальної системи [2]. Цей клас ПЗ є необхідною приналежністю комп'ютера, оскільки забезпечує взаємодію людини, всіх пристроїв та програм комп'ютера. СПЗ координує роботу різних компонентів комп'ютера та відіграє роль посередника між прикладними програмами та апаратним забезпеченням [3]. Чим досконалішим є СПЗ, тим комфортніше почуватися користувач у системному середовищі [4].

До характеристик ПЗ належать: вартість ПЗ, захист ПЗ, повнота реалізації вимог, обсяг файлів ПЗ, вимоги до системного програмного забезпечення та технічних засобів, обсяг потрібної оперативної та дискової пам'яті. Але найважливішими характеристиками як прикладного, так і системного ПЗ з точки зору розробника є його складність, а з точки зору користувача - його якість.

Криза у галузі забезпечення якості системного ПЗ існує - великі проекти виконуються з відставанням від графіка або з перевищенням кошторису витрат, розроблені продукти не мають необхідних функціональних можливостей, продуктивність ПЗ низька, якість ПЗ не влаштовує споживачів. За наближеними оцінками витрати на розроблення ПЗ складають близько 275 мільярдів доларів, але лише 72% програмних проектів досягають етапу впровадження і всього 26% програмних проектів завершуються успіхом [5], тобто лише 71,5 мільярд доларів витрачається на успішні проекти, а решта 200 мільярдів витрачаються на провальні або незавершені проекти. Одним із критеріїв успішності проектів системного ПЗ є їх складність, а відповідно і вартість. Дослідження Standish Group показало, що проекти вартістю менше 750 тисяч доларів виявляються успішними у 55% випадків, вартістю - від 1 до 2 млн - у 18%, вартістю від 5 до 10 мільйонів - всього у 7% випадків [6].

Програмні проекти часто зазнають невдач через помилки на ранніх етапах життєвого циклу ПЗ [7-9]. Отже, *актуальною задачею* наразі є забезпечення можливості оцінювання якості проекту та прогнозування рівня якості розроблюваного за проектом ПЗ на етапі проектування, що дозволить зменшити витрати на розроблення ПЗ, а то й уникнути ряду катастроф та інцидентів, причини яких були внесені на етапах формулювання вимог та проектування.

Постановка задачі. З результатів аналізу сучасного стану галузі оцінювання якості ПЗ слідує, що перспективним напрямком досліджень є розроблення інтелектуального методу, який аналізуватиме і опрацюватиме результати метричного аналізу етапу проектування та надаватиме оцінку якості проекту та прогноз характеристик якості розроблюваного СПЗ.

1. Результати прояву рівня якості системного програмного забезпечення. Для подальшого вирішення задачі оцінювання та прогнозування якості СПЗ слід проаналізувати причини та наслідки відомих випадків прояву низького рівня якості ПЗ.

Аналізу піддавалось найбільш популярне наразі системне програмне забезпечення - операційні системи Android (різних версій), IOS, Microsoft Windows, а також антивірус McAfee. В результаті дослідження було встановлено наступні проблеми [10-15]:

1) у вересні 2012 року компанія Apple визнала проблеми батареї у iPhone - відбувалась швидка втрата заряду батареї через помилку програмного забезпечення операційної системи IOS 5 [14];

2) аналіз мобільної операційної системи Android 2.0 показав наявність 359 програмних помилок, зокрема 88 помилок з високим ступенем ризику та 271 помилка з середнім ступенем ризику, що є серйозною загрозою для безпеки операційної системи [11];

3) помилкове ПЗ на сервері WGA призвело до невірною оновлення операційної системи Windows у серпні 2007 року - всі екземпляри Windows XP і Vista були відмічені як нелегальні [15];

4) помилка компанії Google у 2012 році в операційній системі Android 4.2 призвела до того, що грудень був відсутній в додатку Люди/Контакти [12];

5) у квітні 2010 року компанія McAfee поширила помилковий антивірус, який через помилки у оновленні вірусної бази №5958 атакував системний процес svchost.exe, що призводило до неперервного перезавантаження комп'ютера [13].

Аналіз помилок програмного забезпечення, внесених на різних етапах життєвого циклу (ЖЦ) ПЗ, які стали причинами інцидентів у системному ПЗ, відображено на рис.1 [10].

Наслідками інцидентів, спричинених вищенаведеними помилками, стали: 1) фінансові збитки та втрата репутації компаніями Apple [14], Google [12], McAfee [13], Microsoft [15]; 2) серйозна загроза для безпеки мобільної ОС Android 2.0 через 359 програмних помилок різного ступеня ризику [11].

Розглянуті приклади ілюструють, що деяких помилок можна було уникнути при ранньому оцінюванні проекту та прогнозуванні якості розроблюваного СПЗ. Отже, оцінювати якість проекту, а також прогнозувати якість розроблюваного за проектом ПЗ дійсно необхідно на ранніх етапах життєвого циклу програмного забезпечення, бажано в кінці етапу проектування, тому що ПЗ набуває високої якості не стільки в результаті комплексного тестування кінцевого продукту, скільки в процесі його розроблення.

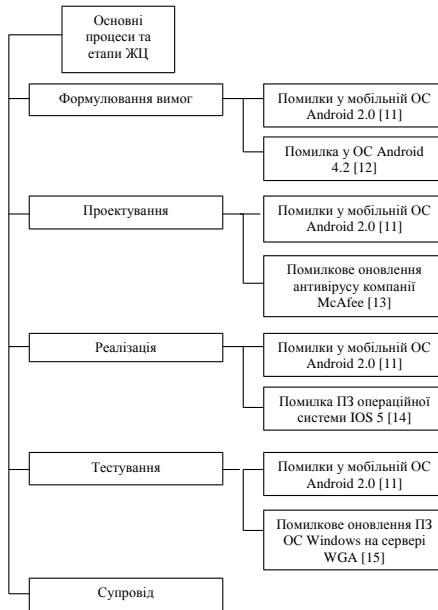


Рис. 1 Помилки у системному ПЗ на різних етапах ЖЦ

2. Проблеми при використанні метрик як інструментів оцінювання характеристик системного програмного забезпечення.

Корисними для оцінювання якості та складності проекту та прогнозування якості та складності розроблюваного за проектом системного ПЗ можуть бути метрики ПЗ. Сучасна програмна індустрія накопичила велику кількість метрик, які оцінюють окремі виробничі та експлуатаційні властивості ПЗ. Однак прагнення їх універсальності, неврахування типу та області застосування розроблюваного ПЗ, ігнорування етапів життєвого циклу ПЗ та необґрунтоване їх використання в процедурах прийняття виробничих рішень істотно підірвало довіру розробників та користувачів ПЗ до метрик. Ці обставини вимагають ретельного відбору метрик для певного типу та області застосування розроблюваного ПЗ, врахування їх обмежень на різних етапах життєвого циклу ПЗ, встановлення порядку їх сумісного використання, накопичення та інтеграції різномірної метричної інформації для прийняття своєчасних виробничих рішень.

При оцінюванні та прогнозуванні характеристик ПЗ на основі аналізу метрик залишається ряд невирішених питань: 1) технологія вимірювання якості ще не досягла зрілості - лише 1.5% софтверних організацій намагаються оцінити якість процесів і готового продукту кількісно, за допомогою метрик, і лише 0.5% софтверних організацій намагаються покращити роботу, керуючись кількісними критеріями якості з метою випуску бездефектних продуктів; 2) відсутні єдині стандарти на метрики - створено більше тисячі метрик, кожен постачальник "вимірювальної" системи пропонує власні способи оцінки якості і відповідно метрики; 3) існує проблема складності інтерпретації величин метрик - значення метрик, одержані за допомогою "вимірювальних систем", неінформативні або малоінформативні для користувача, замовника, а часто і для програміста; 4) метрики розраховуються лише для готового ПЗ - всі "вимірювальні" системи орієнтовані на розрахунок метрик для програмного коду, але часто є необхідність у розрахунку метрик вже на етапі проектування - метрик якості з точними значеннями для проекту ПЗ і метрик якості з прогнозованими значеннями для розроблюваного за проектом програмного забезпечення; 5) низький рівень автоматизації аналізу та опрацювання метрик якості програмного забезпечення - на сьогодні автоматизовано лише процеси збирання, реєстрації та обчислення метричної інформації

Невирішеність цих питань є одним з факторів, що заважає створити бездефектне та високоякісне ПЗ. Складність обґрунтування вибору та інтерпретації метрик в процедурах прийняття виробничих рішень та ігнорування етапів життєвого циклу ПЗ не дозволяють повноцінно використовувати метрики для оцінювання та прогнозування характеристик ПЗ на етапі проектування, а також для підвищення якості ПЗ.

3. Метрична інформація етапу проектування програмного забезпечення. На етапі проектування важливо закласти цілу низку вимог по складності та якості: вимоги до структури програмного забезпечення (ПЗ); вимоги до навігації по ПЗ; вимоги до дизайну інтерфейсів користувача; вимоги до мультимедіа-компонентів ПЗ; вимоги по зручності (usability); технічні вимоги.

На цьому етапі формується відповідь на питання "Яким чином програмна система буде реалізовувати висунуті до неї вимоги?".

Інформаційні потоки етапу проектування ПЗ: вимоги до ПЗ, представлені інформаційною, функційною та поведінковою моделями

аналізу. Інформаційна модель описує інформацію, які повинна обробляти ПЗ на думку замовника. Функційна модель визначає перелік функцій обробки інформації та перелік модулів програмної системи. Поведінкова модель фіксує режими роботи ПЗ. На виході етапу проектування - розробка даних, розробка архітектури та процедурна розробка ПЗ.

З результатів аналізу метрик складності ПЗ та інформаційних потоків етапу проектування ПЗ слідує, що на етапі проектування СПЗ доцільно використати наступні метрики складності, які матимуть точні або прогнозовані значення (таблиця 1) [5, 8].

Таблиця 1. Метрики складності СПЗ етапу проектування

№ п/п	Метрики складності етапу проектування з точними значеннями	Метрики складності етапу проектування з прогнозованими значеннями
1.	Метрика Чепіна	Очікувана LOC-оцінка
2.	Метрика Джилба (абсолютна модульна складність)	Метрика Холстеда
3.	Метрика Мак-Клура	Метрика Маккейба
4.	Метрика Кафура	Метрика Джилба (відносна логічна складність програми)
5.		Прогнозована кількість операторів програми
6.		Прогнозована оцінка складності інтерфейсів ПЗ

Отже, на етапі проектування використаємо 10 основних метрик складності, причому 4 з них мають точні значення, а інші 6 метрик - прогнозовані значення.

Аналіз метрик якості ПЗ та інформаційних потоків етапу проектування ПЗ дозволив зробити висновок, що на етапі проектування застосовні наступні метрики якості з точними або прогнозованими значеннями (таблиця 2) [5, 8].

Таблиця 1. Метрики якості СПЗ етапу проектування

№ п/п	Метрики якості етапу проектування з точними значеннями	Метрики якості етапу проектування з прогнозованими значеннями
1.	Метрика зв'язності	Загальний час розроблення ПЗ (в робочих днях)

2.	Метрика зчеплення	Час етапу проектування (в робочих днях)
3.	Метрика звертання до глобальних змінних	Продуктивність розроблення ПЗ (в хвиликах на один рядок коду)
4.	Час модифікації моделей (в робочих днях)	Вартість перевірки якості ПЗ (в доларах США)
5.	Кількість виявлених помилок при інспектуванні моделей та прототипів	Вартість розроблення ПЗ (в доларах США)
6.		Вартість реалізації коду (в доларах США)
7.		Функційний розмір (FP)
8.		Оцінка трудовитрат за моделлю Боема (в людиномісяцях)
9.		Оцінка тривалості проекту за моделлю Боема (в місяцях)

Отже, на етапі проектування використано 10 основних метрик складності та 14 основних метрик якості програмного забезпечення. Інші метрики є похідними від обраних базових метрик.

Опрацювання вищенаведених 24 метрик складності та якості ПЗ з точними та прогнозованими значеннями є основою для одержання оцінки результатів проектування і прогнозу характеристик складності та якості ПЗ, що розробляється на основі того чи іншого проекту.

4. Нейромережний метод оцінювання результатів проектування і прогнозування складності та якості системного програмного забезпечення (НМОП)

НМОП дозволяє оцінити проект та спрогнозувати характеристики складності та якості розроблюваного ПЗ на основі точних або прогнозованих значень метрик складності та якості ПЗ етапу проектування, перелік яких наведений у попередньому розділі.

НМОП базується на опрацюванні наступних множин: 1) множина метрик складності етапу проектування з точними значеннями $SMEV = \{smev_i | i = 1..4\}$; 2) множина метрик якості етапу проектування з точними значеннями $QMEV = \{qmev_j | j = 1..5\}$; 3) множина метрик складності етапу проектування з прогнозованими

значеннями $CMPV = \{cmpv_k | k = 1..6\}$; 4) множина метрик якості етапу проектування з прогнозованими значеннями $QMPV = \{qmpv_n | n = 1..9\}$.

Результатами опрацювання цих множин є: 1) оцінка складності проекту PCE ; 2) оцінка якості проекту PQE ; 3) прогноз складності розроблюваного за проектом системного програмного забезпечення SCP ; 4) прогноз якості розроблюваного системного програмного забезпечення SQP .

Основою для одержання оцінки складності проекту є елементи множини $CMEV$. Основою для одержання оцінки якості проекту є елементи множин $CMEV$ і $QMEV$. Основою для одержання прогнозу складності розроблюваного ПЗ є елементи множини $CMPV$, але враховуються й елементи множин $CMEV$ і $QMEV$. Основою для одержання прогнозу якості розроблюваного ПЗ є елементи множин $CMPV$ і $QMPV$, але враховуються й елементи множин $CMEV$ і $QMEV$.

Для оцінювання та прогнозування складності та якості ПЗ на основі метричного аналізу слід вирішити задачу визначення взаємозв'язків між значеннями метрик та якістю і складністю проекту та ПЗ. Одним із засобів, який дозволяє узагальнити інформацію та виявити залежності між вхідними і результуючими даними, є штучні нейронні мережі (ШНМ).

Результати метричного аналізу опрацьовуватимемо з використанням ШНМ, яка здійснює апроксимацію метрик ПЗ етапу проектування та надає оцінку складності та якості проекту та прогноз характеристик складності та якості розроблюваного за проектом ПЗ.

НМОП складається з наступних етапів:

- 1) підготовка метрик етапу проектування з точними та прогнозованими значеннями для подання їх на вхід ШНМ;
- 2) перевірка, чи не виходять одержані значення метрик, за межі діапазонів значень входів ШНМ (діапазони вхідних значень ШНМ відображені у [5]);
- 3) опрацювання значень метрик ШНМ;
- 4) аналіз результатів функціонування ШНМ;
- 5) формування висновку про складність та якість проекту і розроблюваного ПЗ на основі результатів ШНМ.

Вхідними даними для ШНМ є множина метрик складності етапу проектування з точними значеннями, множина метрик якості етапу проектування з точними значеннями, множина метрик складності етапу проектування з прогнозованими значеннями, множина метрик якості етапу проектування з прогнозованими значеннями.

Результатом роботи ШНМ є 4 показники: оцінка складності проекту; оцінка якості проекту; прогноз складності розроблюваного за проектом ПЗ; оцінка якості розроблюваного за проектом ПЗ.

У результаті аналізу відомих архітектур штучних нейронних мереж для аналізу метрик етапу проектування ПЗ та прогнозу характеристик якості ПЗ було обрано архітектуру ШНМ - багат шаровий перцептрон.

ШНМ має 9 входів для кількісних значень метрик етапу проектування з точними значеннями та 15 входів для кількісних значень метрик з прогнозованими значеннями. Якщо певна метрика не визначалась, то на відповідний вхід подається -1.

На основі аналізу 4-х одержаних результатів робиться висновок про якість і складність проекту та очікувану якість і складність розроблюваного за проектом системного програмного забезпечення. Оцінка складності проекту, оцінка якості проекту, прогноз складності розроблюваного ПЗ, прогноз якості розроблюваного ПЗ є значеннями з діапазону $[0, 1]$, де 0 - свідчить про недостатність інформації щодо результатів метричного аналізу, близько 0 - проект або розроблюване програмне забезпечення має високу складність або низьку якість відповідно та 1 - проект або розроблюване програмне забезпечення є відповідно простим або високоякісним.

ШНМ має 4 вхідних вектори для подачі значень метрик етапу проектування: на вхід Input1 подаються кількісні значення метрик складності етапу проектування з точними значеннями, на вхід Input2 - кількісні значення метрик якості етапу проектування з точними значеннями, на вхід Input3 - кількісні значення метрик складності етапу проектування з прогнозованими значеннями, на вхід Input4 - кількісні значення метрик якості етапу проектування з прогнозованими значеннями. Граничні значення використовуваних метрик складності та якості наведені у [5].

ШНМ архітектури багат шаровий перцептрон має 24 нейрони вхідного шару, 14 нейронів апроксимуючого шару, 10 нейронів

коригуючого шару та 4 нейрони вихідного шару. Архітектура нейромережної складової НМОП представлена на рис.2.

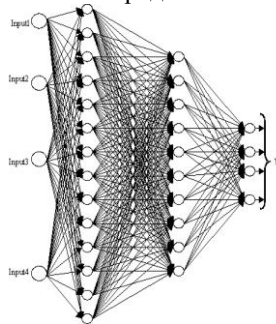


Рис. 2 Архітектура нейромережної складової НМОП

Описану ШНМ реалізовано у пакеті Matlab.

Структурну схему шарів ШНМ у Simulink показано на рис.3.

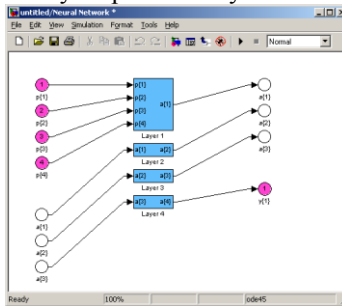


Рис. 3 Структурна схема шарів ШНМ

Вищезазначені залежності результуючих оцінок від вхідних множин метрик враховуються при навчанні нейромережі.

Для навчання одержаної ШНМ послідовність навчальних векторів (навчальна вибірка) задано у вигляді:

$c = \{ [32500; 0; 0; 0] \quad [30875; 0; 0; 0] \quad \dots \};$ - навчальні вектори для входу Input1;

$[0; 0; 0.15; 0; 0] \quad [0; 0; 0.2; 0; 0] \quad \dots \};$ - навчальні вектори для входу Input2;

$[3450; 0; 0; 0; 0; 0] \quad [5900; 0; 0; 0; 0; 0] \quad \dots \};$ - навчальні вектори для входу Input3;

$[0; 0; 0; 0; 1.7; 0; 0; 0; 0] \quad [0; 0; 0; 0; 1.8; 0; 0; 0; 0] \quad \dots \};$ - навчальні вектори для входу Input4.

Цільовий вектор визначено як:

$m = \{ [0.05; 0.02; 0.01; 0.01] \quad [0.1; 0.04; 0.02; 0.02] \dots \}$.

Для вибору алгоритму навчання було проведено навчання ШНМ навчальною вибіркою з 1935 векторів за різними алгоритмами. Для тестування ШНМ використовувалась тестова вибірка з 324 векторів. Дослідження показали, що найменшу похибку навчання одержано при комбінованому критерії якості навчання (msereg); оптимальною кількістю нейронів другого, прихованого, шару є 14 нейронів. Похибка навчання і тестування ШНМ у середньому складає приблизно 0,102197.

Висновки. Надані нейромережним методом оцінювання і прогнозування характеристик програмного забезпечення висновки дозволять оцінити проект, спрогнозувати характеристики розроблюваного за проектом системного програмного забезпечення на основі точних або прогнозованих значень метрик складності та якості ПЗ етапу проектування, а також дозволять порівняти між собою різні версії проектів у ситуації, коли вартість і тривалість приблизно однакові.

Запропонований нейромережний метод оцінювання і прогнозування характеристик ПЗ надає замовнику інформацію для вибору проекту ПЗ та дозволяє порівняти між собою різні версії проекту, тобто є основою для прийняття мотивованого та обгрунтованого рішення щодо вибору проекту та його реалізації з врахуванням характеристик складності та якості проекту і розроблюваного програмного забезпечення.

Використані джерела:

1. Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение – СПб: Питер, 2003 – 736 с.
2. Рошин А.В. Системное программное обеспечение: Учебное пособие – М.: МГУПИ, 2007 – 166 с.
3. Васильев В.Г. Введение в системное программное обеспечение: Учебное пособие – Тверь: ТГТУ, 2009 – 160 с.
4. Журавлева Т.Ю. Системное и прикладное программное обеспечение: Учебное пособие – М.: Изд-во Московского государственного открытого университета, 2010 – 144 с.
5. Мищенко В.О., Поморова О.В., Говорущенко Т.А. CASE-оценка критических программных систем. В 3-х томах. Том 1.

- Качество / Под ред. Харченко В.С. - Харьков: Нац.аэрокосмический университет "ХАИ", 2012. - 201 с.
6. William J. Brown, Raphael C. Malveau, Hays W. McCormick, Thomas J. Mowbray. AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis - Wiley, 1998 - 336 p.
 7. С.Макконнелл. Совершенный код. Мастер-класс - М.: Издательство "Русская редакция", 2013 - 896 с.
 8. O.Pomorova, T.Hovorushchenko. Intelligent Assessment and Prediction of Software Characteristics at the Design Stage // American Journal of Software Engineering and Applications, 2013; 2(2) - pp. 25-31 // [Electronic resource] – Access mode: <http://article.sciencepublishinggroup.com/pdf/10.11648.j.ajsea.20130202.11.pdf>
 9. Поморова О.В., Говорущенко Т.О. Сучасні проблеми оцінювання якості програмного забезпечення // Радіоелектронні і комп'ютерні системи – Харків: НАУ “ХАІ”, 2013 – № 5, с.319-327
 10. Т.О.Говорущенко, Р.А.Малярчук, І.В.Вовк. Аналіз інцидентів, спричинених помилками програмного забезпечення // Інтелектуальні технології в системному програмуванні. II Всеукраїнська науково-практична конференція молодих учених та студентів. Збірник наукових праць - Хмельницький: Тріада-М, 2013 - с.187-198
 11. Study: 359 Android code flaws pose security risks // [Electronic resource] – Access mode: http://news.cnet.com/8301-30685_3-20021437-264.html?part=rss&subj=news&tag=2547-1_3-0-20
 12. Google promises fix for Android 4.2 December birthday bug // [Electronic resource] – Access mode: <http://www.androidcentral.com/google-s-promises-fix-android-42-december-birthday-bug>
 13. McAfee DAT 5958 Update Issues // [Electronic resource] – Access mode: <http://isc.sans.edu/diary/McAfee+DAT+5958+Update+Issues/8656>
 14. How to fix battery life issues with iOS 6 or iPhone 5 // [Electronic resource] – Access mode: <http://www.imore.com/how-fix-battery-life-problems-ios-6-or-iphone-5>
 15. MS server error marks PCs as ‘nongenuine’ // [Electronic resource] – Access mode: <https://windowssecrets.com/patch-watch/ms-server-error-marks-pcs-as-nongenuine/>