

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Науковий вісник НЛТУ України
Scientific Bulletin of UNFU<https://nv.nltu.edu.ua><https://doi.org/10.36930/40300219>

Article received 29.04.2020 p.

Article accepted 04.06.2020 p.

UDC 004.[052+416.2]



ISSN 1994-7836 (print)

ISSN 2519-2477 (online)

@ ✉ Correspondence author

B. V. Uhrynovskiy

bohdanuhryn@gmail.com

В. С. Яковина, Б. В. Угриновський

Національний університет "Львівська політехніка", м. Львів, Україна,

**СТАРІННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОБІЛЬНИХ ДОДАТКІВ:
АНАЛІЗ ПРОБЛЕМАТИКИ**

Виконано огляд та аналіз літературних джерел, в яких досліджено явища старіння програмного забезпечення мобільних додатків. Визначено основні характеристики явища старіння програмного забезпечення. Встановлено, що мобільні системи та додатки є особливо вразливі до ефектів старіння і потребують детальних досліджень. Охарактеризовано основні методи та засоби дослідження явища старіння, що застосовуються для його вивчення в мобільній системі Android. Описано загальну схему дослідження явища старіння, яка дає змогу проводити експерименти та визначати наявність чи відсутність старіння в системі, а також вказує на вплив факторів на прояви старіння. Визначено використовувані індикатори старіння, а саме такі індикатори системи та додатків, як тривалість запуску Android Activity, оперативна пам'ять, файлове сховище, використання CPU, Garbage Collector. Виділено основні фактори, що впливають на прояви явища старіння: технічні характеристики пристрою, типи додатків та програмний код, інтенсивність запуску додатків, події введення, оперативна пам'ять та пам'ять файлового сховища. Встановлено, що згідно з результатами попередніх досліджень, ефективними алгоритмами машинного навчання для визначення наявності старіння є метод опорних векторів та дерева прийняття рішень. Проаналізовано наявні дослідження, методи та засоби виконання процедури омолодження програмного забезпечення для зменшення впливу старіння на надійність системи Android. З'ясовано, що для протидії старінню програмного забезпечення в мобільній системі Android пропонують засоби як на рівні розроблення архітектури та реалізації мобільного додатку, так і на системному рівні і рівні компонент. Встановлено, що ключовим засобом протидії старінню є перезавантаження компонент на рівні системи (наприклад, Activity manager) чи додатків (Java-контейнери), а також є необхідність розроблення таких засобів для планування виконання процедури омолодження. Обґрунтовано актуальність впливу явища старіння на забезпечення надійності сучасних мобільних та вбудованих систем. Визначено напрями майбутніх досліджень, а саме: визначення ефективних факторів і індикаторів для мобільних систем, побудова моделей старіння, розроблення методів і засобів омолодження програмного забезпечення мобільних систем.

Ключові слова: надійність; фактори старіння; індикатори старіння; модель старіння; модель омолодження; ланцюг Маркова.

Вступ. Явище старіння програмного забезпечення – це процес накопичення помилок у стані системи впродовж усього часу виконання, що призводить до погіршення продуктивності та виникнення відмов програмного забезпечення [19]. Старіння програмного забезпечення має кумулятивний характер, тому особливо часто проявляється в системах, які працюють тривалий час без перезавантаження. Причинами старіння є помилки, які можуть накопичуватись в системі, але проявлятися з часом, наприклад, витоки пам'яті. Чинники, що спричиняють виникнення помилок старіння, називають факторами старіння [1, 19]. Старіння програмного забезпечення призводить до погіршення характеристик надійності [26], які водночас є критичними властивостями більшості сучасної техніки загалом і програмного забезпечення зокрема. Одним з важливих показників на-

дійності є *час до відмови* внаслідок старіння або ж *час до виснаження*, коли йдеться про використання ресурсів системи [13].

У попередній роботі [30] визначено характеристики явища старіння програмного забезпечення і зроблено висновок, що мобільні та вбудовані пристрої є особливо вразливі до проявів ефектів старіння. Старіння програмного забезпечення проявляється не тільки в системному [17] або прикладному програмному забезпеченні персональних комп'ютерів чи серверів [20], але і в мобільних додатках таких систем, як Android [12; 18]. Мобільні пристрої, на відміну від більшості персональних комп'ютерів, безперервно використовуються впродовж тривалого часу без перезавантаження. Отже, мобільні пристрої є особливо вразливими до появи старіння програмного забезпечення. Старіння програмного за-

Інформація про авторів:

Яковина Віталій Степанович, д-р техн. наук, професор, завідувач кафедри програмного забезпечення.

Email: vitaliy.s.yakovyna@lpnu.ua; <https://orcid.org/0000-0003-0133-8591>

Угриновський Богдан Володимирович, аспірант, кафедра програмного забезпечення. **Email:** bohdanuhryn@gmail.com

Цитування за ДСТУ: Яковина В. С., Угриновський Б. В. Старіння програмного забезпечення мобільних додатків: аналіз проблематики. Науковий вісник НЛТУ України. 2020, т. 30, № 2. С. 107–112.

Citation APA: Yakovyna, V. S., & Uhrynovskiy, B. V. (2020). Software aging in the context of reliability: a review of the issue. *Scientific Bulletin of UNFU*, 30(2), 107–112. <https://doi.org/10.36930/40300219>

безпечення, внаслідок накопичення помилок у стані системи і збільшення витрат ресурсів, таких як фізична пам'ять, призводить до погіршення продуктивності пристрою, збільшення затримки відгуку інтерфейсу і, врешті-решт, збоїв у роботі додатку чи відмов системи виконувати свої функції, що має значний вплив на задоволення користувачів та позитивний досвід використання системи [28].

Відкрита мобільна платформа Android на базі ядра операційної системи Linux є найпопулярнішою мобільною системою і станом на 2020 р. становить 86,6 % [9] ринку мобільних систем. Хоча система Android є Linux-системою, неможливо повністю повторювати схеми досліджень, які використовувались для системи Linux. Наприклад, простір підкачування є важливим показником у виявленні старіння програмного забезпечення Linux, але він використовується в такому вигляді в системі Android [27].

Об'єктом дослідження є явище старіння програмного забезпечення.

Предметом дослідження є методи і засоби, які дають змогу визначити фактори та індикатори старіння програмного забезпечення, а також побудувати відповідні моделі його старіння та омолодження.

Мета роботи – огляд та аналіз факторів, індикаторів та методів дослідження, а також моделей старіння та омолодження програмного забезпечення для операційної системи Android, оскільки дослідження процесу старіння ПЗ вбудованих і мобільних операційних систем перебувають на ранній стадії.

Завдання дослідження: опис та характеристика особливостей методології дослідження явища старіння програмного забезпечення в операційній системі Android; визначення та аналіз факторів і індикаторів старіння; огляд та аналіз моделей старіння та омолодження; виявлення перспективних напрямів майбутніх досліджень старіння програмного забезпечення мобільних операційних систем.

Наукова новизна результатів дослідження цієї статті полягає у тому, що на підставі аналізу моделей методів та засобів вивчення явища старіння програмного забезпечення визначено основні характеристики явища старіння програмного забезпечення для операційної системи Android та розроблено напрями майбутніх досліджень щодо омолодження програмного забезпечення для зменшення впливу старіння на надійність системи Android.

Практична значущість результатів дослідження полягає в описі та виявленні особливостей експериментального підходу дослідження явища старіння програмного забезпечення на платформі Android, а також у висвітленні практичних задач майбутніх досліджень.

Аналіз літературних джерел. Дослідження характеристик, факторів та індикаторів старіння платформи Android [12, 16, 18, 23] описують особливості процесу старіння та визначають ті індикатори, які давали б змогу найточніше виявляти наявність цього процесу. Такими індикаторами та факторами є тривалість процесу запуску Android Activity [6], використання оперативної пам'яті процесами, індикатори роботи Garbage collector, час роботи процесів та використання CPU. Розробляють та досліджують моделі старіння та омолодження для протидії явищу старіння програмного забезпечення з метою подальшої реалізації стратегій та алгоритмів моні-

торингу, планування та виконання процедури омолодження. Для операційної системи Android пропонують підходи омолодження як на рівні операційної системи [18, 29], шляхом моніторингу системних індикаторів та планування виконання процедури омолодження, так і з допомогою алгоритмів та рекомендацій для мінімізації робочого навантаження на мобільний пристрій [28].

Фактори та індикатори старіння. Дослідження факторів, як чинників та джерела проявів старіння ПЗ, є початковим етапом вивчення явища старіння для конкретних платформ та середовищ. Для Android платформи розроблено методології та підходи для вимірювання індикаторів та визначення факторів старіння. Роботи [12, 16, 18] реалізують методології виявлення та аналізу процесу старіння програмного забезпечення в мобільній системі Android, в яких можна виділити шість основних кроків проведення дослідження (рисунок).



Рисунок. Загальна схема дослідження явища старіння програмного забезпечення в операційній системі Android

На початковому етапі визначається стратегія моніторингу, тобто вибираються метрики та індикатори системи, компонент чи додатків, які будуть відстежуватись, а також реалізуються технічні засоби для проведення вимірювань. Метрики для вимірювання, які були використані в роботах [12, 16, 18], подано в табл. 1. Індикатори можуть прямо чи опосередковано вказувати на наявність ефектів старіння. Наприклад, збільшення тривалості запуску Android Activity вказує на погіршення продуктивності, що може відчувати користувач безпосередньо під час взаємодії з пристроєм. Водночас, збільшення використання оперативної пам'яті чи дискового простору не обов'язково означатиме погіршення продуктивності чи збільшення кількості відмов.

Табл. 1. Індикатори старіння програмного забезпечення в операційній системі Android

Метрика	Опис
Тривалість запуску Android Activity	Тривалість процесу запуску екранів додатку від моменту ініціалізації до відображення на екрані.
Оперативна пам'ять system_server PSS; Free Memory; Cached Memory; Lost RAM; Used PSS; ZRAMinSWAP; ZRAMPhysicalUsed.	Глобальні метрики пам'яті, використання пам'яті процесами.
Файлове сховище	Глобальні метрики пам'яті, метрики читання і запису.
Прибиральник сміття Garbage collector pause time (GC-paused); total time (GC-total).	Паралельний та явний збирач сміття. Причини та тривалість затримок.
Процеси та CPU	Час роботи процесів та помилки сторінки.

У роботі [12, 16, 18] збір інформації відбувається з допомогою комп'ютера та програмної утиліти adb [5], яка контролює мобільний пристрій Android через USB зв'язок. Для збирання даних використовуються деякі утиліти Linux, що доступні в системі Android, наприклад, команди `top` [14], `iostat`. Також для отримання даних доступні команди Android SDK, такі як `logcat` [8], `dumpsys` [7]. Ще одним джерелом даних є файлова система ядра Linux `proc`, яка забезпечує доступ до файлів, які містять інформацію про використання пам'яті, дискового простору, планування та пам'ять процесів.

Наступним кроком у дослідженні ефектів та факторів старіння є розроблення алгоритму генерування робочого навантаження [13] для реалізації стресового тестування та імітації тривалого використання мобільного пристрою. В роботі [12, 16, 18] використано утиліту Monkey [10], яка дає змогу генерувати події користувачького введення, навігації, запусків Activity [4] тощо.

Далі, ґрунтуючись на обраних стратегіях моніторингу та на генеруванні робочого навантаження, розробляється план виконання стресового тестування. Вибір факторів, що можуть впливати на ефекти старіння, є важливим аспектом під час розроблення дослідження та планування експериментів [16]. Фактори, як і вимірювані індикатори, можуть мати прямий чи опосередкований вплив на прояви ефектів старіння. Наприклад, конфігурація пристрою визначає обмеження ресурсів пам'яті та CPU, що може мати безпосередній вплив на погіршення продуктивності, а робоче навантаження в різних випадках може по-різному активувати процес старіння. Під час побудови плану тестування враховуються різні комбінації факторів, забезпечуючи оптимальний час виконання тесту (табл. 2).

Табл. 2. Фактори старіння програмного забезпечення в операційній системі Android

Фактор	Опис
Мобільний пристрій	Модель виробника та технічні характеристики
Мобільний додаток	Розробник, технологія, складність та категорія програмного забезпечення
Запуск і завершення додатків	Інтенсивність запусків і повторних запусків додатків
Події введення	Перемикання додатків, навігація по додатку, події сенсорного вводу
Об'єм оперативної пам'яті	Використання та витрати пам'яті
Об'єм файлового сховища	Об'єм вільного простору

У роботі [12] запропоновано підхід до дослідження процесу старіння ПЗ системи Android з допомогою програмних утиліт для збирання інформації (Linux команда `top`) про використання пам'яті та генерування стресового навантаження (інструмент Monkey Android середовища), а також виконано аналіз можливого впливу застосованих утиліт на результати виконаних експериментів. Результати аналізу використання оперативної пам'яті процесами для збирання інформації вказують на ефективність проведення досліджень таким методом, оскільки ці процеси мають незначний вплив на використання пам'яті.

Отримані результати стресових тестів та експериментів аналізуються з допомогою статистичних методів, щоб виявити наявність явища старіння та визначити вплив факторів на нього. У роботах [16, 18] представлено розгорнуту експериментальну методологію

для аналізу проблем старіння програмного забезпечення в операційній системі Android. Методологія використовує статистичні методи, щоб визначити, які фактори (такі як робочі навантаження та конфігурації пристроїв) посилюють погіршення продуктивності та споживання ресурсів. Окрім цього, методологія аналізує взаємозв'язок між старінням програмного забезпечення та показниками використання ресурсів, щоб визначити, які підсистеми впливають на старіння та забезпечити розроблення стратегій омолодження програмного забезпечення. Емпіричний аналіз старіння програмного забезпечення з допомогою розробленої методології показав, що [16]:

- 1) Старіння програмного забезпечення дійсно впливає на пристрої Android. Після декількох годин стрес-тестування пристрої відчули помітне погіршення продуктивності (з точки зору збільшення затримки відгуку системи). На продуктивність впливають робоче навантаження (програми, події) і конфігурація (зокрема, доступний об'єм пам'яті для зберігання даних).
- 2) Проблеми, пов'язані зі старінням програмного забезпечення, можуть бути пов'язані з конкретними процесами в операційній системі Android, включаючи System Server, System UI і Surface Flinger, які демонструють завищені витрати пам'яті. Окрім цього, специфічні сервіси всередині них, такі як Activity Manager та Power Manager, демонструють тенденції до старіння.

Тенденції до погіршення продуктивності корелюють з показниками використання ресурсів ядра, такі як PSS процесу і сторінок, об'єднаних з допомогою KSM, і з часом, витраченим на збір сміття. Ці індикатори використані [18] як ознаки старіння програмного забезпечення в комбінації із тривалістю запуску Android Activity і визначені як критерії з високим рівнем довіри.

Деякі алгоритми машинного навчання використовували для вивчення характеристик старіння програмного забезпечення, такі як дерево прийняття рішень (Decision Tree – ID3) [2, 3, 21, 24, 26], метод опорних векторів (Support Vector Machine – SVM) [11], нейронні мережі [24], і так далі. У роботі [23] розроблено експерименти для порівняння ефективності трьох алгоритмів машинного навчання: дерево рішень, метод опорних векторів, глибинна мережа переконань (Deep Belief Network – DBN), а також впроваджено новий показник Page fault для виявлення старіння програмного забезпечення в системі Android. Порівняння ефективності трьох алгоритмів машинного навчання показало, що дерево рішень і SVM є більш ефективними, ніж DBN. Індикатор помилки сторінки Page fault в експериментах показав, що стабільність моделей з використанням цього індикатора старіння є трохи кращою за ті моделі, що використовують тривалість запуску Activity для виявлення старіння програмного забезпечення в системі Android, що означає, що він також може бути використаний як хороший індикатор старіння Android.

Моделі старіння та омолодження програмного забезпечення. Окрім дослідження ефектів, факторів та індикаторів старіння, для Android платформи також пропонують різні підходи до зменшення впливу старіння та використання технік омолодження програмного забезпечення.

Щоб протидіяти старінню програмного забезпечення в мобільних пристроях, в статті [28, 30] пропонують новий алгоритм розбиття додатків для розвантаження,

який має на меті розділити програму на локальні (обчислення виконуються безпосередньо на мобільному пристрої) та віддалені (наприклад, виконання складних обчислень відбувається на віддаленому сервері) частини, забезпечуючи при цьому загальну вартість як можна меншою. Загалом, розвантаження є корисним, якщо віддалене виконання має кращу продуктивність, ніж виконання на пристрої, або, що рівнозначно, якщо витрати на передачу даних на віддалений сервер менші, ніж вираш у часі або використання енергії під час локального виконання обчислень. В роботі розглядаються тільки дві задачі оптимізації, хоча цей алгоритм розбиття може використовуватися для інших цілей оптимізації. Схема часткового розвантаження здатна ефективно зменшити загальну тривалість виконання програми, а також споживання енергії. Окрім цього, вона може певною мірою адаптуватися до змін у середовищі та уникнути різкого зниження продуктивності додатків після різкого падіння пропускної здатності. Тому під час запуску деяких складних додатків протягом тривалого періоду продуктивність системи не буде різко погіршуватися, а стабільність пристрою гарантується шляхом уникнення старіння програмного забезпечення.

Для Android побудовано модель омолодження програмного забезпечення [29], яка ґрунтується на моделі поведінки використання телефону у вигляді ланцюга Маркова, що складається з двох станів (використання та очікування) для того, щоб можна було передбачити поведінку користувача у всіх часових інтервалах дня. Процеси старіння з омолодженням і без омолодження моделюються як два окремі ланцюги Маркова відповідно [22]. Використовуючи стохастичну мережу Петрі (SPN) і інструмент Oris [15, 25], було згенеровано інтегрований безперервний часовий ланцюг Маркова (Continuous Time Markov Chain – CTMC), який базується на підставі моделі індивідуальної поведінки використання та моделі процесу старіння. Завдяки інтегрованій моделі CTMC можна виконувати попередню стратегію активного омолодження, враховуючи як режим використання, так і стан старіння телефону Android. Оптимальний інтервал омолодження може бути визначений шляхом максимізації доходу та мінімізації витрат. У роботі використовують марковські моделі переважно з метою спрощення аналізу, однак, пропонується використовувати і інші немарківські моделі з загальними розподілами, які можуть більш вдало відобразити поведінку користувача та процес старіння.

У роботі [16] реалізовано агент виявлення старіння та омолодження в ОС Android під назвою ADARTA (Aging Detection And Rejuvenation Tool for Android). Архітектура агент складається з трьох основних компонентів: монітори навантаження та ключових показників ефективності; детектор старіння; модуль омолодження. Модуль омолодження використовує інформацію, яку надає детектор старіння (рівень довіри, час до відмови старіння (Time To Aging-Failure – TTAF) та рівень навантаження), щоб вирішити, коли застосувати омолодження системи. Наприклад, налаштування модуля можуть викликати негайне омолодження, якщо TTAF короткий незалежно від поточного навантаження, або якщо TTAF довший, омолодження може бути відкладено до часу, коли поточне навантаження буде низьким. Омолодження проводиться шляхом перезавантаження ключових служб, наприклад, таких як WiFi service та

Activity manager. Також пропонують реалізацію точнішого методу омолодження шляхом очищення окремих Java контейнерів (наприклад, ArrayList, HashMap) в межах обраних критичних служб процесу системного сервера. Основна ідея цього підходу – це профілювання процесів і служб для розуміння того, як розвивається використання пам'яті об'єктів. Далі застосовують деякі евристики для виявлення та очищення контейнерів Java, не викликаючи побічних ефектів у програмі. Використані евристики можуть вказувати тільки на контейнери, розмір яких зростає з часу або контейнери, які мають, щонайменше, один об'єкт з тривалим терміном служби.

Обговорення результатів дослідження. Описана схема дослідження явища старіння в мобільній системі Android може бути використана і адаптована під особливі потреби в майбутніх роботах. Цей підхід можливо застосувати і для інших мобільних систем, зокрема iOS, для якої на цей час немає досліджень явища старіння програмного забезпечення.

Виявлені і описані метрики та фактори старіння програмного забезпечення в системі Android потребують подальшого детального вивчення, оскільки важливо вибрати валідні і ефективні індикатори для виявлення старіння, а також визначити фактори, що впливають на процес старіння. Варто зауважити, що більшість робіт і досліджень зосереджені на вивченні явища старіння на рівні операційної системи, не враховуючи вплив користувацьких додатків та динаміку їх старіння. Наприклад, необхідно визначити індикатори, які найбільш повно характеризують погіршення продуктивності роботи додатків і збільшення затримки зворотного відгуку системи. Деякі індикатори можуть давати недостатньо інформації про стан старіння в певних сценаріях використання мобільного пристрою чи додатків. Зокрема, індикатор тривалості запуску Android Activity обмежений в тому, що він дає змогу оцінити тривалість запуску екрану додатку тільки один раз під час запуску додатку і не відображає динаміку зміни продуктивності під час виконання цього додатку. При цьому потрібно враховувати використання ресурсів системи (наприклад, пам'ять) та роботу системних процесів (збирач сміття, фонові процеси відповідальні за роботу сенсорів та інші).

Важливо розглянути і вивчити вплив інших факторів на процес старіння. Виявлені фактори важливі як для розуміння процесу старіння загалом, так і для побудови ефективних алгоритмів протидії старінню. Варто розглянути різні сценарії використання мобільного пристрою користувачем в якості фактора, наприклад, тривале використання одного додатку без його перезавантаження може мати істотні відмінності від сценарію з частим перемиканням між додатками. Також можна розглядати такі фактори, як тип мобільного додатку, архітектура додатку, технічні характеристики мобільного пристрою та версії операційної системи.

Аналіз алгоритмів і моделей старіння та омолодження програмного забезпечення дає підстави зробити висновок про те, що дослідження перебувають на ранній стадії, оскільки більшість робіт описують теоретичні моделі, а експерименти мають певні обмеження в застосуванні, однак розглянуті дослідження мають перспективу бути вдосконаленими та реалізованими. Наприклад, модель омолодження [29], яка враховує модель по-

ведінки користувача, потребує додаткових досліджень, особливо важливо перевірити її ефективність на реальних тестових даних. Також цю модель можна розширити, додавши нові фактори, наприклад, фактор стану заряду батареї. Відповідно реалізований агент ADARTA [18] для виявлення старіння та омолодження в ОС Android може бути застосовано для перевірки інших метрик та факторів старіння. Також у майбутніх роботах варто розглядати комбінації методів вимірювання системних індикаторів та теоретичних моделей омолодження для реалізації ефективних механізмів виявлення старіння, планування і виконання процедури омолодження.

Висновок. Сучасні мобільні та вбудовані пристрої вразливі до впливів явища старіння програмного забезпечення, оскільки вони часто мають обмежені ресурси та тривалий час працюють без перезавантаження. Тенденція до забезпечення високих показників якості та надійності мобільних пристроїв вимагає детального вивчення явища старіння та розроблення методів та засобів запобігання його проявів.

Дослідження процесу старіння програмного забезпечення в мобільних системах перебувають на ранній стадії і стосуються тільки системи Android. Використання крос-платформових технологій та різноманітних архітектурних рішень під час розроблення додатків можуть бути факторами, які впливають на прояви старіння та погіршення характеристик надійності загалом. Необхідне подальше дослідження таких факторів, а також дослідження явища старіння ПЗ на інших мобільних операційних системах.

У роботі визначено фактори (технічні характеристики пристрою, типи додатків та програмний код, інтенсивність запуску додатків, події введення, оперативна пам'ять та пам'ять файлового сховища) та індикатори (тривалість запуску Android Activity, використання оперативної пам'яті та файлового сховища, використання CPU, робота Garbage Collector) для виявлення та боротьби з ефектами старіння програмного забезпечення. Ці фактори і індикатори потребують подальших досліджень для отримання точних оцінок їх впливу на процес старіння програмного забезпечення операційної системи Android. Також, доцільним було б впровадження нових індикаторів (наприклад, повідомлення Application Not Responding чи профілювання відображення інтерфейсу) та факторів (наприклад, типи мобільних додатків, архітектура та алгоритми додатків, конфігурації пристрою та операційної системи) для більш точного опису процесів старіння програмного забезпечення мобільних систем.

Дослідження та застосування засобів омолодження програмного забезпечення можливе як на рівні розроблення архітектури та реалізації мобільного додатку, так і на системному рівні і рівні компонент. Дослідження та реалізація засобів для вимірювання наявності явища старіння програмного забезпечення в мобільній системі та застосування механізмів перезапуску служб та окремих компонент сервісів системи є важливим напрямом майбутніх досліджень. Також можливим завданням наступних досліджень є розроблення рекомендацій для протидії ефектам старіння, зокрема розроблення алгоритмів зменшення використання локальних ресурсів мобільного пристрою.

References

1. Abdullah, Z. H., Yahaya, J. H., Mansor, Z., & Deraman, A. (2017). Software Ageing Prevention from Software Maintenance Perspective – A Review. *Journal of Telecommunication, Electronic and Computer Engineering*, 9(3–4), 93–96.
2. Alonso, J., Belanche, L., & Avresky, D. (2011). Predicting software anomalies using machine learning techniques. *Proceedings of 2011 IEEE International Symposium on Network Computing and Applications*, 163–170. <https://doi.org/10.1109/NCA.2011.29>
3. Alonso, J., Torres, J., Berral, J. L., & Gavalda, R. (2010). Adaptive on-line software aging prediction based on machine learning. *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, 507–516. <https://doi.org/10.1109/DSN.2010.5544275>
4. Android. (2020). *Activity – Android Developers*. January 29. Retrieved from: <https://developer.android.com/reference/android/app/Activity>
5. Android. (2020). *Android Debug Bridge*. January 29. Retrieved from <https://developer.android.com/studio/command-line/adb>
6. Android. (2020). *App startup time – Android Developers*, May 5. Retrieved from: <https://developer.android.com/topic/performance/vitals/launch-time>
7. Android. (2020). *Dumpsys – Android Developers*. January 29. Retrieved from: <https://developer.android.com/studio/command-line/dumpsys>
8. Android. (2020). *Logcat command-line tool*, January 29. Retrieved from: <https://developer.android.com/studio/command-line/logcat>
9. Android. (2020). *Smartphone OS market share*, January 29. Retrieved from: <http://www.idc.com/promo/smartphone-market-share/os>
10. Android. (2020). *UI/Application Exerciser Monkey*, January 29. Retrieved from: <https://developer.android.com/studio/test/monkey>
11. Andrzejak, A., & Silva, L. (2008). Using machine learning for non-intrusive modeling and prediction of software aging. *NOMS 2008-2008 IEEE Network Operations and Management Symposium*, 25–32. <https://doi.org/10.1109/NOMS.2008.4575113>
12. Araujo, J., Alves, V., Oliveira, D., Dias, P., Silva, B., & Maciel, P. (2013). An Investigative Approach to Software Aging in Android Applications. *2013 IEEE International Conference on Systems, Man, and Cybernetics*. <https://doi.org/10.1109/SMC.2013.213>
13. Bao, Y., Sun, X., & Trivedi, K. S. (2005). A workload-based analysis of software aging and rejuvenation. *IEEE Transactions on Reliability*, 54(3), 541–548. <https://doi.org/10.1109/TR.2005.853442>
14. Blum, R. (2008). *Linux Command Line and Shell Scripting Bible*. Indianapolis, IN: Wiley Publishing, Inc.
15. Bucci, G., Camevali, L., Ridi, L., & Vicario, E. (2010) Oris: a tool for modeling, verification and evaluation of real-time systems. *International Journal on Software Tools for Technology Transfer*, 12(5), 391–403. <https://doi.org/10.1007/s10009-010-0156-8>
16. Cotroneo, D., Fucci, F., Iannillo, A. K., Natella, R., & Pietrantuono, R. (2016). Software aging analysis of the android mobile os. *IEEE 27th International Symposium on Software Reliability Engineering*, 478–489. <https://doi.org/10.1109/ISSRE.2016.25>
17. Cotroneo, D., Natella, R., Pietrantuono, R., & Russo, S. (2010). Software aging analysis of the linux operating system. *2010 IEEE 21st International Symposium on Software Reliability Engineering*. <https://doi.org/10.1109/ISSRE.2010.24>
18. Cotroneo, D., Simone, L. D., Natella, R., Pietrantuono, R., & Russo, S. (2019). A Configurable Software Aging Detection and Rejuvenation Agent for Android. *11th International Workshop on Software Aging and Rejuvenation (WoSAR)*. <https://doi.org/10.1109/ISSREW.2019.00078>
19. Grottke, M., Jr, R. M., & Trivedi, K. S. (2008). The fundamentals of software aging. *IEEE International Conference on Software Reliability Engineering Workshops*, 1–6. <https://doi.org/10.1109/ISSREW.2008.5355512>

20. Grottke, M., Li, L., Vaidyanathan, K., & Trivedi, K. S. (2006). Analysis of software aging in a web server. *IEEE Transactions on Reliability*, 55(3), 411–420. <https://doi.org/10.1109/TR.2006.879609>
21. Hayashi, T., & Ohta, S. (2014). Performance degradation detection of virtual machines via passive measurement and machine learning. *International Journal of Adaptive Resilient and Autonomic Systems*, 5(2), 40–56. https://doi.org/10.4018/978-1-5225-1759-7_ch086
22. Huang, Y., Kintala, C., Kolettis, N., & Fulton, N. (1995). Software rejuvenation: analysis, module and applications. *Proceedings of Twenty-Fifth International Symposium on Fault-Tolerant Computing*, 381–390. <https://doi.org/10.1109/FTCS.1995.466961>
23. Huo, S., Zhao, D., Liu, X., Xiang, J., Zhong, Y., & Yu, H. (2018). Using machine learning for software aging detection in Android system. *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*. <https://doi.org/10.1109/ICACI.2018.8377553>
24. Magalhaes, J., & Silva, L. (2010). Prediction of performance anomalies in web-applications based-on software aging scenarios. *2010 IEEE Second International Workshop on Software Aging and Rejuvenation*. <https://doi.org/10.1109/WOSAR.2010.5722095>
25. ORIS. (2020). Tool Analysis of timed and stochastic Petri nets, January 29 Retrieved from: <https://www.oris-tool.org/>
26. Polovko, A. M., & Gurov, S. V. (2008). *Fundamentals of Reliability Theory*. St. Petersburg: BHV-Petersburg, 704. [In Russian].
27. Qiao, Y., Zheng, Z., & Qin, F. Y. (2016). An empirical study of software aging manifestation in android. *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 84–90. <https://doi.org/10.1109/ISSREW.2016.19>
28. Wu, H., & Wolter, K. (2015). Software aging in mobile devices: Partial computation offloading as a solution. *2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 125–131. <https://doi.org/10.1109/ISSREW.2015.7392057>
29. Xianga, J., Wenga, C., Zhao, D., Tiana, J., Xiong, S., Lia, L., & Andrzejak, A. (2019). A New Software Rejuvenation Model for Android. *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. <https://doi.org/10.1109/ISSREW.2018.00021>
30. Yakovyna, V. S., & Uhrynovskiy, B. V. (2019). Starinnia prohrannoho zabezpechennia v konteksti yoho nadiinosti: ohliad problematyky. *Scientific Bulletin of UNFU*, 29(5), 123–128. <https://doi.org/10.15421/40290525>

V. S. Yakovyna, B. V. Uhrynovskiy

Lviv Polytechnic National University, Lviv, Ukraine

SOFTWARE AGING IN THE CONTEXT OF RELIABILITY: A REVIEW OF THE ISSUE

Software aging is the process of errors accumulation in the state of system throughout the runtime that results in performance degradation and increase of failure rate. This paper carried out the review and analysis of literary sources devoted to the study of the software aging phenomenon in mobile operating systems, such as Android. Mobile devices run for a long time without restarting and users often switch applications. Therefore, mobile devices are vulnerable to the effects of aging, because conditions of use may result in accumulation of errors in the system, such as memory leaks. The study described a general algorithm for detecting and investigating the effects of aging in the Android mobile system. The main metrics and factors used in studies are identified. The paper highlighted main metrics that indicate the presence of software aging in Android operating system, namely memory indicators like PSS (Proportional Set Size), Garbage Collector metrics and Activity launch time. This work also described main factors that influence the effects of software aging: device configurations, application types and source code, application launch and input events rate, amount of usable and free RAM and data storage, memory leaks. Detection of software aging occurs by measuring system metrics taking into account external and internal factors and by applying statistical methods (alternatively, machine learning methods can be used). The purpose of aging effects detecting and factors and metrics investigation is to schedule and perform a rejuvenation procedure. Software rejuvenation is a technique to prevent and delay software aging. The paper analyzed approaches of software rejuvenation in the Android mobile operating system. The main approach to counteracting software aging phenomena is active rejuvenation, which consists in scheduling of reboots of system components at different levels under certain conditions and optimal time. For example, at the system level services such as Activity manager can be restarted, and at the level of individual process Java containers (ArrayList, HashMap) can be cleaned. As a result, the paper identified directions of future research, namely: identifying of effective factors and metrics for mobile systems, building software aging models, developing methods and tools for software rejuvenation in mobile operating system.

Keywords: software reliability; aging factors; aging metrics; aging model; rejuvenation model; Markov chain.