ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «СИСТЕМА АВТОМАТИЗАЦІЇ ОБСЛУГОВУВАННЯ ПРИСТРОЇВ В УМОВАХ ВІДДАЛЕНОЇ РОБОТИ»

на здобуття освітнього ступеня магістра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Владислав ГАФАНОВИЧ*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач(ка) вищої освіти гр. КСДМ-62
Владислав ГАФАНОВИЧ
Ім'я, ПРІЗВИЩЕ
Керівник: Людмила АСЄЄВА
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент:
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти Магістр

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Наталія ЛАЩЕВСЬКА
_____Ім'я, ПРІЗВИЩЕ
«_____» _____ 2024р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гафанович Владислав Олександрович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Система автоматизації обслуговування пристроїв в умовах віддаленої роботи»
Людмила АСЄЄВА, phD (доктор філософії), доцент ,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024р. №320,
2. Строк подання кваліфікаційної роботи «29» грудня 2024р.
3. Вихідні дані до кваліфікаційної роботи: Науково-технічна література, стандарти та протоколи для управління пристроями, технологічні тенденції в автоматизації, вимоги до обслуговування розподіленої системи, практичні випадки використання та дані для розробки сценаріїв (скриптів), економічні та організаційні фактори.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Автоматизація технічного обслуговування кінцевих пристроїв
2. Методи і засоби автоматизації технічного обслуговування кінцевих пристроїв
3. Розробка та впровадження системи автоматизованого технічного обслуговування
5. Перелік ілюстративного матеріалу: *презентація*
   1. Мета, об'єкт, предмет дослідження
   2. Актуальність роботи
   3. Проблема та необхідність автоматизації обслуговування пристроїв у віддаленій роботі
   4. Процес впровадження автоматизації у віддаленому обслуговуванні
   5. Автоматизація обслуговування через сценарії (скрипти)
   6. Технічні вимоги та залежності для сценаріїв (скриптів)
   7. Покрокова демонстрація роботи сценарію (скрипта) безперервного розгортання ПЗ
   8. Фрагмент скрипта та виклики підпрограм
   9. Отримані результати від впровадження автоматизованого рішення
   10. Порівняння запропонованого рішення
   11. Висновки

12. Продовження висновків
13. Публікації та апробація роботи

6. Дата видачі завдання «01» вересня 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	10.10-12.10.2024	
2	Обробка матеріалу	12.10-15.10.2024	
3	Розробка теоретичної частини	15.10-18.10.2024	
4	Методи і засоби автоматизації технічного обслуговування кінцевих пристроїв	18.10-22.10.2024	
5	Розробка та впровадження системи автоматизованого технічного обслуговування	22.10-26.11.2024	
6	Висновки за результатами аналізу	29.11-30.11.2024	
7	Розробка презентації	30.11-08.12.2024	
8	Передзахист	09.12-11.12.2024	
9	Здача в деканат	20.12-29.12.2024	

Здобувач вищої освіти _____ Владислав ГАФАНОВИЧ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Людмила АСЄЄВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістр: 148 стор., 83 рис., 2 табл., 46 джерел.

Мета роботи – розробити та дослідити систему автоматизації обслуговування пристроїв для забезпечення ефективного управління та моніторингу в умовах віддаленої роботи, з акцентом на зниження витрат і підвищення надійності.

Об'єкт дослідження – процес обслуговування та управління пристроями на віддалених робочих місцях.

Предмет дослідження – засоби та методи автоматизації обслуговування пристроїв в умовах віддаленої роботи.

Короткий зміст роботи: У роботі аналізуються основні аспекти автоматизації обслуговування пристроїв для віддалених працівників, зокрема роль хмарних рішень, автоматизованих оновлень і діагностики. Окрему увагу приділено питанням безпеки, автономності та масштабованості таких систем, що дозволяє ефективно управляти інфраструктурою без постійної залежності від інтернет-з'єднання.

Запропоновані підходи оптимізують процеси обслуговування за допомогою скриптів і автоматичних оновлень, що мінімізують втручання операторів і знижують витрати на підтримку пристроїв. Розроблена система також демонструє економічний ефект завдяки скороченню витрат на ручне обслуговування і зниженню потреби в додаткових ресурсах.

КЛЮЧОВІ СЛОВА: АВТОМАТИЗАЦІЯ, РОЗПОДІЛЕНЕ ОБСЛУГОВУВАННЯ ПРИСТРОЇВ, КІНЦЕВІ ПРИСТРОЇ, ВІДДАЛЕНЕ УПРАВЛІННЯ, ХМАРНІ ТЕХНОЛОГІЇ, ІНТЕРНЕТ РЕЧЕЙ (IOT), МОНІТОРИНГ ПРИСТРОЇВ, УПРАВЛІННЯ ІНФРАСТРУКТУРОЮ, БЕЗПЕКА СИСТЕМ, АВТОМАТИЗОВАНІ РІШЕННЯ, СЦЕНАРІЇ АВТОМАТИЗАЦІЇ.

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 148 pages, 83 figures, 2 tables, 46 sources.

The purpose of the work is to develop and research a device maintenance automation system to ensure effective management and monitoring in remote work environments, with an emphasis on reducing costs and increasing reliability.

The object of the study is the means and methods of automating the maintenance of devices in conditions of remote work.

The subject of the research is the means and methods of automating the maintenance of devices in conditions of remote work.

Summary of the work: The work analyzes the main aspects of automating the maintenance of devices for remote workers, in particular, the role of cloud solutions, automated updates and diagnostics. Particular attention is paid to the issues of security, autonomy and scalability of such systems, which allows efficient management of the infrastructure without constant dependence on an Internet connection.

The proposed approaches optimize maintenance processes with the help of scripts and automatic updates, which minimize operator intervention and reduce device maintenance costs. The developed system also demonstrates an economic effect due to the reduction of manual maintenance costs and the reduction of the need for additional resources.

KEY WORDS: AUTOMATION, DISTRIBUTED DEVICE MAINTENANCE, END DEVICES, REMOTE CONTROL, CLOUD TECHNOLOGIES, INTERNET OF THINGS (IOT), DEVICE MONITORING, INFRASTRUCTURE MANAGEMENT, SYSTEM SECURITY, AUTOMATED SOLUTIONS, AUTOMATION SCENARIOS.

ЗМІСТ

ВСТУП

Світ перебуває у стрімкому переході до моделі віддаленої роботи, яка задовільняє потреби більшості користувачів. Завдяки цьому працівники можуть виконувати свої обов’язки без необхідності фізично знаходитися в офісі, маючи лише ноутбук та доступ до інтернету. Однак для підтримання безперебійної роботи таких систем потрібні значні ресурси для ефективного управління та технічного обслуговування великої кількості кінцевих пристроїв, що розташовані в різних куточках світу. З розвитком технологій і впровадженням розподілених інфраструктур зростає потреба в автоматизованих рішеннях, які здатні забезпечити стабільну роботу віддалених систем.

Це питання стає особливо актуальним для великих підприємств та організацій, де значна кількість працівник працює поза офісом, використовуючи корпоративні пристрої. Традиційне ручне технічне обслуговування в таких умовах стає малоефективним і вимагає значних витрат часу та ресурсів. Потрібно постійно контролювати кожен окремий пристрій, що створює додаткове навантаження на технічний персонал і значно збільшує витрати. Автоматизація цього процесу дозволяє не лише суттєво знизити витрати на технічне обслуговування, але й підвищити швидкість реагування на можливі збої, мінімізувати участь людського фактора та зменшити кількість помилок.

Завдяки впровадженню автоматизованих систем можна підвищити надійність кінцевих пристроїв і забезпечити більш точний моніторинг у реальному часі. Дана робота спрямована на дослідження та розробку систем автоматизації технічного обслуговування в умовах розподілених мереж з віддаленим доступом, які використовують наймані віддалені працівники. Метою є створення ефективних рішень, які оптимізують процеси обслуговування, зменшують витрати і підвищують дієвість

РОЗДІЛ 1 ОГЛЯД АВТОМАТИЗАЦІЇ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ КІНЦЕВИХ ПРИСТРОЇВ

1.1 Необхідність автоматизації технічного обслуговування в умовах віддаленої роботи

З розвитком сучасного суспільства ми все більше відходимо від традиційної моделі, де щоденна робота була пов’язана з необхідністю фізично перебувати в офісі або на робочому місці протягом усього дня. Сьогодні, під впливом різних факторів, зокрема зростання мобільності працівників і розширення кадрового штату, стало можливим працювати віддалено. Це рішення особливо актуальне у випадках, коли компанія стикається з обмеженням офісних приміщень або коли найманим працівникам немає потреби перебувати в офісі для виконання своїх обов’язків. Роботодавці все частіше визнають, що працівники можуть ефективно виконувати свої завдання, не потребуючи постійного контролю, якщо для цього створено всі необхідні умови.

Глобальні події, що вплинули на традиційні умови праці, змусили багатьох перейти на віддалений формат роботи, що стосується не лише трудової діяльності, а й навчання. Віддалене навчання може викликати певний скептицизм, особливо серед тих, хто лише починає свій шлях у професії, оскільки вони не завжди мають достатньо практичного досвіду. Деякі роботодавці ставляться насторожено до випускників, які здобули більшу частину своєї освіти дистанційно, адже навчання і робота мають різні вимоги. Присутність у навчальному середовищі важлива для повноцінного навчання, особливо на ранніх етапах. Це пов’язано з тим, що не всі технології, такі як VR і AR, ще широко доступні і застосовні. Для тих, хто вже має певний досвід роботи або тільки розпочинає свою кар’єру, дистанційна робота пропонує безліч переваг. Вони можуть налаштувати робочий простір у себе вдома, створюючи умови, що сприяють підвищенню продуктивності, і уникати щоденних

поїздок до офісу. Хоча в умовах віддаленої роботи інколи можуть виникати складнощі через зовнішні подразники, все більше компаній визнають, що переваги гнучкого графіку і можливість організувати власне робоче середовище переважують ці незначні незручності.

Віддалена робота, насправді, не є чимось новим. Значна кількість людей вже багато років успішно працює у даному форматі, ще задовго до останніх глобальних подій. Це явище було розповсюджене, однак останнім часом воно набуло особливої популярності, оскільки роботодавці дедалі частіше усвідомлюють переваги віддаленої роботи. Компанії почали розуміти, що не обов'язково орендувати великі офіси для того, щоб забезпечити робочі місця для всіх працівників. Частина працівників може працювати в невеликому центральному офісі, а інші виконуватимуть свої обов'язки віддалено. У деяких випадках всі працівники можуть повністю працювати з дому, що дозволяє значно оптимізувати витрати на офісні приміщення. Цей підхід також відповідає сучасним вимогам ринку праці, де дистанційна робота вже давно стала звичним явищем для багатьох професіоналів. Є приклади людей, які десятиліттями працюють віддалено і при цьому досягають високих результатів. Їм не потрібно витрачати час і кошти на поїздки до офісу, що дозволяє їм зосередитися на своїх завданнях і підтримувати високий рівень продуктивності. Вони отримують таку ж зарплату, як і працівники в офісі, але при цьому економлять на транспортних витратах і часі, який можна використати більш ефективно.

Віддалена робота допомагає людині раціональніше організовувати свій робочий день – коли закінчується робочий час, вона просто вимикає ноутбук або комп'ютер і переходить до особистих справ, не витрачаючи додатковий час на дорогу додому. Такі працівники, як правило, виглядають менш втомленими і емоційно виснаженими, оскільки у них відсутні стреси, пов'язані з щоденними поїздками. Натомість працівники, які змушені їздити до офісу, часто страждають від перевтоми, навіть якщо їхня робота не є надто складною. Довгі поїздки туди й назад забирають у них чимало енергії та часу, що призводить до того, що вони приходять додому виснаженими. Часто у таких людей залишається лише декілька

годин на те, щоб повечеряти, переглянути щось для відпочинку і лягти спати, щоб на наступний день повторити цей цикл, що може призвести до емоційного вигорання і психологічного виснаження, навіть якщо робота сама по собі не є важкою. Особливо складною є ситуація для тих, хто живе далеко від робочого місця. Якщо до стресу від роботи додається ще й фізична втома від щоденних поїздок, то рано чи пізно така людина може вирішити залишити роботу на користь більш перспективної і менш виснажливої.

Значне зростання числа працівників, які працюють віддалено, відбулося внаслідок подій, що розгорнулися на початку цього десятиліття. Ми наблизилися до етапу, коли велика частина роботи, що раніше виконувалася в офісах, може бути успішно виконана з дому. У майбутньому це стане ще більш поширеною практикою. Адже багато завдань, особливо тих, які пов’язані з роботою за комп’ютером, не вимагають фізичної присутності в офісі. Виникає логічне запитання: для чого працівнику їхати до офісу, щоб працювати за тим же комп’ютером, якщо можна зробити це з дому? Навіть якщо є необхідність роздруковувати документи або забезпечити їх підписання, сучасні технології швидко змінюють ці процеси, зводячи бюрократичні бар’єри до мінімуму.

Світ вже давно усвідомив необхідність зменшення використання обмежених ресурсів, таких як папір, пластик, і мінімізації шкідливих викидів в атмосферу. Наразі йде активна робота над тим, щоб зробити всі процеси більш енергоефективними, з переходом на акумулятори та інші інноваційні рішення, що дозволить забезпечити краще майбутнє для наступних поколінь. Технології рухаються вперед, і це не тільки про екологію, а й про зручність у робочих процесах, що дозволяє уникати зайвих витрат часу.

Що стосується бюрократичних перепон, то в нашій країні вже йде активна цифрова трансформація. Менше часу витрачається на фізичні відвідини установ для підписання документів або здійснення різних процедур. Багато процесів уже автоматизовані. Наприклад, у випадку з оплатою комунальних послуг у багатьох країнах це вже автоматично знімається з рахунків банків, що спрощує життя громадян. Якщо у когось виникає заборгованість, то вона накопичується не перед

державою, а перед банком. Таким чином, люди не змушені виконувати зайві дії, що економить їх час та зусилля. Хоча ще є велика кількість людей, особливо похилого віку, які традиційно відвідують поштові відділення для здійснення таких операцій, поступово ці процеси переходять до автоматизації. Наприклад, квитанції все ще видаються вручну, але це вже більше виняток, ніж правило. Сучасні банки та системи вже готові до автоматичного оброблення платежів, проте перехід до повної автоматизації поки що не завершено. Але цей напрямок розвитку є очевидним і невідворотним. Цей приклад наочно демонструє, що світ постійно змінюється, і ми, як суспільство, змушені трансформуватися разом із ним.

Технології стають невід'ємною частиною нашого життя, і важливо не лише приймати їх, але й розуміти та ефективно використовувати. Неправильно вважати, що певні види робіт не можуть виконуватися віддалено. Так, деякі роботи, такі як ремонт техніки, вимагають присутності спеціалістів на місці, але навіть ці професії можуть бути адаптовані до віддаленого виконання за наявності відповідного обладнання в домі працівника. Наприклад, фахівець з ремонту, маючи необхідні інструменти вдома, може виконувати більшість завдань, не виходячи з дому, що економить час і ресурси. Якщо говорити про будівельні професії, то поки що ці фахівці не можуть працювати віддалено, оскільки процес будівництва вимагає фізичної присутності. Однак навіть у цій сфері вже є приклади інновацій, як використання 3D-принтерів для будівництва будинків, що вже активно впроваджується в Китаї, доводячи, що автоматизація може охопити навіть такі традиційні галузі, як будівництво.

Більшість видів робіт можна автоматизувати до такого рівня, коли працівники будуть виконувати лише аналітичні або інтелектуальні завдання, працюючи з даними, без необхідності витрачати час на поїздки до офісу чи робочого місця. Цей підхід відкриває безліч можливостей для працівників: вони можуть працювати з будь-якого місця, де є інтернет-зв'язок і необхідне обладнання. Вони можуть поїхати на відпочинок, взявши з собою ноутбук, і продовжувати працювати, не перериваючи робочий процес. Автоматизація йде швидкими кроками, і віддалена робота вже стала реальністю, яка приносить значні

переваги порівняно з традиційними робочими умовами. Вона дозволяє людям ефективніше використовувати свій час, знижуючи витрати на поїздки та інші супутні незручності, і пропонує більшу гнучкість у виборі робочого місця.

Тепер слід звернути увагу на дещо інше, як ефективно організувати роботу та підтримувати працівників, які працюють віддалено, особливо коли вони стикаються з технічними проблемами. Не кожен з них може самостійно вирішити такі проблеми, і тому виникає потреба у спеціальних рішеннях для віддаленої підтримки.

Сьогодні існують досить прості інструменти, такі як TeamViewer або AnyDesk, які дозволяють дистанційно підключатися до комп'ютера та виконувати необхідні налаштування. Ці програми надають змогу допомогти користувачам з будь-якої точки світу, навіть з мобільного пристрою. Проте, такі рішення можуть бути незручними, коли йдеться про велику кількість працівників.

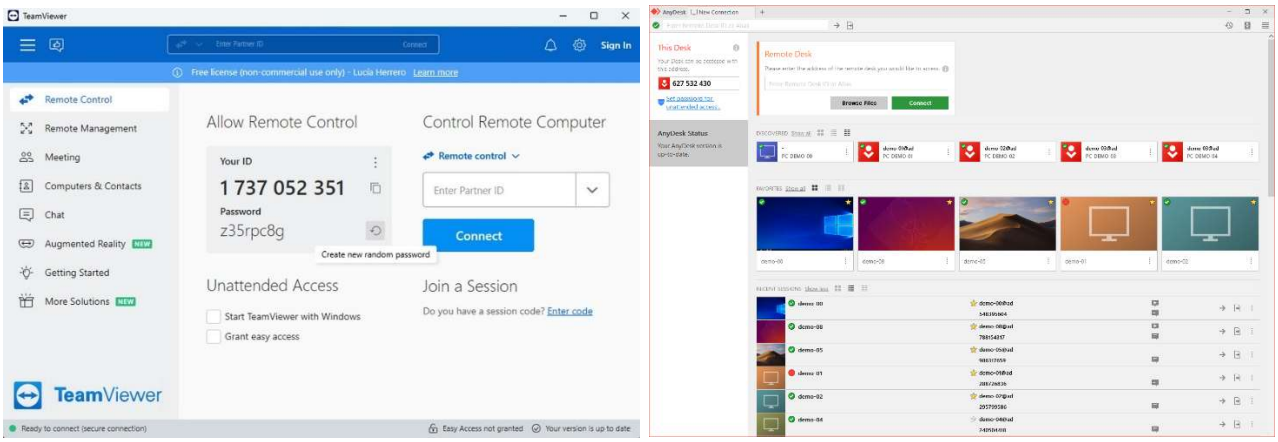

Рисунок 1.1 – ПЗ для віддаленої підтримки TeamViewer та AnyDesk
(не адаптований для підтримки великої кількості пристроїв;
в разі використання у комерційних цілях – блокування)

Коли йдеться про кілька осіб, наприклад, один або два працівники, використання TeamViewer є цілком прийнятним. У випадку великого штату, який може нараховувати десятки або сотні працівників, що працюють у різних точках світу, підключатися до кожного окремо стає непродуктивним. Виникає ризик зупинити їх роботу під час виконання налаштувань або вирішення технічних

питань, що негативно вплине на загальну ефективність компанії. Навіть для простих технічних завдань, таких як налаштування програмного забезпечення чи оновлення системи, може знадобитися певний час, і це може зайняти від години до двох, залежно від складності задачі. А якщо таких працівників не декілька, а десятки або сотні, процес стає ще більш ускладненим.

Великі компанії, де працівники розкидані по різних локаціях, важливо знайти рішення, яке дозволить надавати технічну підтримку без затримок і перешкод для основної роботи працівників. Існує необхідність забезпечити доступ до всіх працівників у реальному часі, при цьому не заважаючи їхній роботі та не створюючи додаткових витрат для компанії. Наприклад, коли потрібно виконати оновлення програмного забезпечення або вирішити технічну проблему, важливо зробити це максимально швидко та ефективно, щоб не переривати робочий процес і не створювати простоїв. Завдання має на меті надання інструментів та системи підтримки, які нададуть своєчасне реагування на технічні проблеми, зберігаючи продуктивність і безперебійність робочих процесів.

Коли ми говоримо про автоматизацію, варто згадати, як за допомогою технологій можна значно спростити та масштабувати підтримку великої кількості робочих місць, зокрема тих, що знаходяться в одному приміщенні. Замість того, щоб фізично підходити до кожного комп’ютера для налаштування, ми можемо використовувати автоматизовані сценарії або скрипти, тобто заздалегідь підготовлені набори команд, які виконують конкретні дії, які нагадують своєю будовою на кшталт програмування. Однак, на відміну від програмування, скрипти орієнтовані більше на операційні завдання, тобто виконання конкретних дій у системі.

Автоматизація дозволяє уникнути багатьох незручностей, які виникають під час оновлення чи налаштування системи, зокрема з боку кінцевих користувачів. Використовуючи скрипти, ми можемо виконувати потрібні завдання у фоновому режимі, тобто без необхідності втручання працівників у процес, тобто коли проводиться оновлення або налаштування, особи, які користуються комп’ютерами, інші не помічають цих змін. Всі ці дії виконуються автоматично, без появи

інтерактивних вікон чи інших відволікаючих елементів, що дозволяє працівникам безперебійно продовжувати роботу. Як приклад, після виконання певної автоматизованої операції, скрипт може надіслати працівнику повідомлення з проханням, наприклад, змінити пароль. При цьому працівник просто отримує повідомлення про необхідність ввести новий пароль під час наступного входу в систему, без детальних пояснень про те, які саме зміни відбулися, що значно спрощує робочий процес, оскільки працівникам не доводиться турбуватися про технічні деталі чи процеси, що відбуваються за лаштунками, вони просто виконують інструкції і продовжують працювати далі.

Такий підхід до автоматизації дозволяє не лише підвищити ефективність роботи, а й знизити кількість помилок, адже дії виконуються за заздалегідь підготовленими сценаріями. До того ж це мінімізує час на технічне обслуговування, адже адміністратору не потрібно особисто втручатися в кожен процес на кожному робочому місці — все виконується одночасно і без зайвих затримок. Цей приклад показує, як робота з віддаленим доступом може бути значно простішою, коли підтримка надається для конкретного фізичного місця, де працюють інші працівники.

Існує багато варіантів автоматизації, які вже активно використовуються для полегшення таких процесів. Деякі адміністратори створюють прості команди або невеликі скрипти для вирішення локальних задач, тоді як інші розробляють складні, масштабні сценарії. Ці сценарії можуть включати різноманітні параметри та змінні, що дозволяють адаптувати автоматизацію до конкретних пристроїв або середовищ, забезпечуючи більш плавну роботу.

Такі складні скрипти надають бажане, бо дозволяють зменшити кількість необхідних ручних втручань, оскільки вже заздалегідь враховують різноманітні нюанси. Автоматизовані сценарії часто містять змінні, які легко підлаштовуються під різні умови, що дозволяє використовувати їх універсально для великої кількості пристроїв або робочих станцій. Але, навіть із такою гнучкістю, підтримка та регулярне оновлення таких скриптів залишаються важливим аспектом роботи.

1.2 Основні поняття та характеристики кінцевих пристроїв у розподілених системах

Якщо раніше можна було говорити про актуальність автоматизації технічного обслуговування в умовах віддаленої роботи, то зараз варто приділити особливу увагу поняттю і характеристикам кінцевих пристроїв у розподілених системах. У сучасних умовах ми стикаємося з тим, що працівники можуть підключатися до корпоративних систем з різних місць і з використанням різноманітних пристроїв або обладнання, надане роботодавцем, так і особисті комп'ютери або ноутбуки працівників, кожен з яких має свої технічні особливості, які впливають на продуктивність та швидкодію (див. рис.1.2).

Рисунок 1.2 – Різноманіття кінцевих пристроїв у розподілених системах (пристрої, що використовують віддалені працівники)

Основний виклик заключається саме в забезпеченні можливісті віддаленого обслуговування цих пристроїв, що є дуже складним завданням через різноманітність пристроїв і конфігурацій (див. рис.1.1). Наприклад, працівники можуть користуватися обладнанням з різним рівнем продуктивності, що безпосередньо впливає на ефективність їхньої роботи та технічну підтримку.

Важливо не забувати, що для забезпечення безперебійної роботи, працівники повинні мати стабільний доступ до Інтернету (див. рис.1.2) [1]. Хоча це може бути не найшвидше підключення, воно повинно бути достатньо надійним для забезпечення безперебійної взаємодії між працівником і технічною підтримкою, особливо під час віддаленого технічного обслуговування.

Рисунок 1.3 – Складність мережевої інфраструктури під час віддаленої роботи

Технічні спеціалісти стикаються з додатковими проблемами, оскільки їм доводиться мати справу з великою кількістю різних пристроїв. Кожен пристрій має неподібний до інших, унікальний функціонал, що вимагає не тільки глибоких технічних знань, а й гнучкості в підході до вирішення проблем, які виникають в процесі віддаленого обслуговування. (див. рис.1.2, 1.3) [1].

Кінцевий пристрій – це апаратний або програмний компонент, який слугує інтерфейсом між користувачем та системою. У розподілених системах кінцеві пристрої виступають точками доступу до мережі або систем, через які відбувається взаємодія з іншими вузлами або ресурсами. Ці пристрої можуть мати різні форми: від стаціонарних персональних комп’ютерів та ноутбуків до мобільних пристроїв, серверів або навіть пристроїв Інтернету речей, таких як принтери, камери, сенсори тощо (див. рис.1.2).

Кінцеві пристрої охоплюють значно більше, ніж просто систему, яку використовує працівник для виконання своєї роботи, наприклад, ноутбук . Є безліч інших пристроїв, з якими він може взаємодіяти і які можуть бути необхідними для

виконання його завдань таких як принтери, камери або сенсори, надані компанією для конкретних проектів, наприклад, для збору даних або проведення тестувань. Такі пристрої, як правило, підключаються до мережі Інтернет, що дозволяє здійснювати їх інтеграцію з основною системою. Якщо базова система, наприклад, комп'ютер, підключена до тієї ж мережі, це відкриває можливість для проведення технічного обслуговування. Досить важливо, що це обслуговування може бути автоматизованим, що значно спрощує підтримку таких пристроїв на віддалених робочих місцях. Автоматизація дозволяє оперативно вирішувати технічні питання без необхідності фізичної присутності спеціалістів, забезпечуючи стабільну роботу кінцевих пристроїв та безперебійний доступ до ресурсів і мереж (див. рис.1.2, 1.3) [1].

Існує декілька основних типів кінцевих пристроїв, які використовуються у віддаленій роботі (див. рис.1.4) [2, 3]:

— ПК або ноутбуки;

— мобільні пристрої.

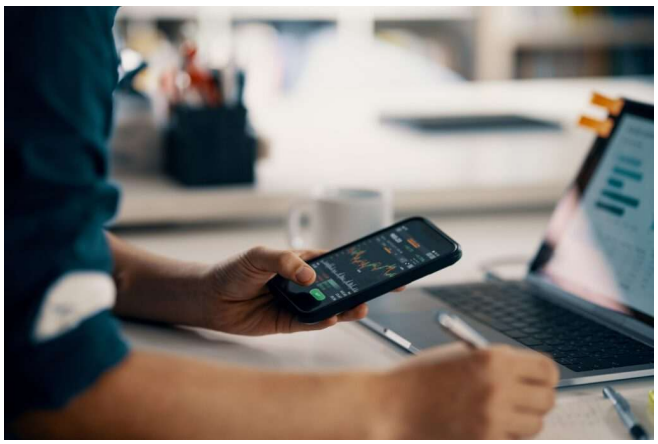

Рисунок 1.4 – Основні кінцеві пристрої, які використовуються на віддаленній роботі (ПК або ноутбуки та мобільні пристрої)

ПК або ноутбуки можуть бути власністю працівника, придбаними ним для виконання робочих завдань, або наданими роботодавцем чи компанією для корпоративних цілей. У випадку корпоративних рішень роботодавець може надавати працівникам спеціальні пристрої, щоб вони використовували їх виключно

для роботи і не застосовували особисті комп’ютери для доступу до корпоративних систем, що забезпечить додаткову безпеку і контроль, адже доступ до корпоративних ресурсів здійснюється лише з конкретних пристроїв, виділених компанією. У віддаленій роботі такі комп’ютери та ноутбуки є основним інструментом для виконання робочих завдань, надаючи працівникам можливість безпечного доступу до систем та ресурсів (див. рис.1.4) [2, 3].

Другий тип кінцевих пристроїв – це мобільні пристрої, зокрема смартфони та планшети. Вони також активно використовуються під час віддаленої роботи, особливо для комунікації та доступу до корпоративних програм та інструментів. Мобільні пристрої можуть використовуватися для електронної пошти, відеоконференцій, обміну миттєвими повідомленнями, доступу до хмарних сервісів тощо. Однак технічне обслуговування таких пристроїв потребує особливих підходів, зокрема управління безпекою та оновлення мобільних операційних систем – це може включати налаштування акаунтів (наприклад, Google або Apple iCloud), що дозволяє контролювати доступ і забезпечувати безпеку даних. Також часто компанії надають працівникам корпоративні SIM-карти для використання на мобільних пристроях, за допомогою яких можна здійснювати дзвінки та обмін даними в рамках єдиного корпоративного контракту (див. рис.1.5) [4].

Такий підхід забезпечує контроль над витратами на зв’язок і може бути важливим для тих працівників, чия робота пов’язана з частими телефонними дзвінками або іншими комунікаційними завданнями, окрім роботи з документами та корпоративними системами (див. рис.1.4) [2].

Рисунок 1.5 – Важкість у технічному обслуговуванні кінцевих пристроїв, що використовуються віддаленими працівниками

Найменш поширеними, але набираючий велику популярність серед кінцевих пристроїв, які можуть в майбутньому використовуватися у віддаленій роботі, є пристрої Інтернету речей (IoT) – це такі різноманітні датчики, розумні пристрої, відеокамери та інші подібні технології, які також можуть бути частиною великої розподіленої системи. Такі пристрої, хоча поки не використовуються у віддаленій роботі в порівнянні з персональними комп'ютерами чи ноутбуками, все ж можуть відігравати важливу роль у певних майбутніх проектах, особливо у випадках, коли необхідно виконати збір даних або забезпечити моніторинг (див. рис.1.6) [5].

Рисунок 1.6 – Різноманіття пристроїв Інтернету речей (IoT)

Пристрої Інтернету речей потребують специфічного технічного обслуговування та моніторингу через їхні обмежені обчислювальні можливості. Наприклад, обслуговування таких пристроїв залежить від стандартів безпровідного зв'язку, за якими вони працюють. Якщо пристрій використовує Wi-Fi, то обмеження на обмін даними не настільки значні, як у випадку з технологіями Zigbee або Z-Wave – ці технології, хоч і є більш енергоефективними, мають дуже низьку пропускну здатність – до 240 кбіт/с, що для простих датчиків достатньо, адже вони не потребують великого обсягу даних (див. рис.1.5) [5]. Проте, для складніших пристроїв, таких як відеокамери, які обробляють та передають відео, така низька пропускна здатність виявиться недостатньою.

Пристрої Інтернету речей ще не набули актуальності у віддаленій роботі, але вони можуть використовуватися в окремих випадках, особливо для спеціалізованих завдань. Наприклад, вони можуть бути частиною проєктів, де потрібно розробляти чи тестувати датчики для підприємств або інших замовників. У таких ситуаціях працівники можуть використовувати персональні комп'ютери чи ноутбуки для взаємодії з цими IoT-пристроями, розробляючи, налаштовуючи або віддалено керуючи ними [6].

Основна робота у віддаленій системі виконується за допомогою традиційних пристроїв, таких як комп'ютери та ноутбуки, а Інтернет речей частіше використовується у вузьких і спеціалізованих випадках, де необхідний моніторинг, збір даних або інші подібні процеси, що включають взаємодію з датчиками чи розумними пристроями [5, 6].

Досить важливим і розповсюдженим типом кінцевих пристроїв у сучасних системах можна вважати віртуальні машини та хмарні ресурси (див. рис.1.6) [7]. Багато компаній або володіють власними серверами, або орендують їх, або ж використовують корпоративні рішення на основі підписок на хмарні сервіси. У цьому контексті, кінцевими пристроями в розподілених системах можуть бути саме віртуальні машини, які функціонують у хмарних середовищах або віртуалізованих інфраструктурах. Їхнє технічне обслуговування охоплює управління ресурсами та забезпечення безпеки в хмарних середовищах, що є все більш популярним

рішенням для багатьох компаній, але використання таких ресурсів залежить від політики самої компанії. Якщо організація застосовує потужності хмарних сервісів або віртуалізованих систем, працівники можуть підключатися до віртуальних машин, що знаходяться на серверах компанії або орендованих нею ресурсах (див. рис.1.5, 1.7) [4, 7].

Рисунок 1.7 – Віртуалізовані машини та віддалені робочі столи

Віртуальні машини дають можливість працівникам працювати навіть на застарілому обладнанні, так як основні обчислювальні процеси відбуваються на сервері, а не на локальному пристрої, що дозволяє працівникам віддалено підключатися до потужностей серверів компанії, за умови наявності стабільного та швидкого інтернет-з'єднання. Такий підхід є вигідним у випадках, коли компанія не надає персональні комп'ютери або ноутбуки, а замість цього забезпечує доступ до віртуальних ресурсів. Працівник може виконувати завдання на віртуальній машині, яка знаходиться на сервері компанії, і таким чином отримувати доступ до високопродуктивних обчислювальних ресурсів незалежно від якості його особистого обладнання. Навіть якщо працівник має застарілий комп'ютер, підключившись до хмарних ресурсів або віртуальної машини, він може користуватися всіма перевагами сучасних корпоративних рішень (див. рис. 1.7) [7].

Автоматизація технічного обслуговування таких систем часто є простішою, ніж обслуговування розподілених систем, що базуються на різних ноутбуках і персональних комп'ютерах. Оскільки всі обчислювальні потужності та ресурси

знаходяться в одному місці — на сервері, це спрощує управління, моніторинг і підтримку системи. У випадку використання віртуальних машин, автоматизація дозволяє централізовано контролювати і підтримувати роботу серверів, що значно полегшує процеси технічного обслуговування в порівнянні з підтримкою розподілених кінцевих пристроїв, які фізично знаходяться в різних місцях. Віддалена робота у такому контексті дозволяє працівникам ефективно взаємодіяти з основними робочими інструментами, такими як комп'ютери або ноутбуки, при цьому використовуючи потужні ресурси віртуальних машин. Це дає їм можливість виконувати більш складні завдання без необхідності мати на руках сучасне обладнання, оскільки всі ресурси, необхідні для продуктивної роботи, надаються хмарними або віртуалізованими рішеннями (див. рис. 1.7) [7].

Якщо говорити про основні характеристики кінцевих пристроїв, то варто виділити кілька ключових аспектів:

— мобільність;

— безпека;

— оновлення та підтримка ПЗ.

Мобільність. У контексті віддаленої роботи більшість кінцевих пристроїв є мобільними, зокрема ноутбуки та смартфони. Мобільність надає користувачам свободу переміщатися і працювати з будь-якого місця, що є великою перевагою для багатьох працівників. Наприклад, для тих, хто активно користується смартфонами для дзвінків чи організації роботи через ноутбук, можливість пересуватися є зручним фактором (див. рис.1.4) [2, 3].

Однак для технічного обслуговування це створює певні труднощі, оскільки такі пристрої часто змінюють своє фізичне розташування і мережеві підключення. Не завжди можна передбачити, чи має працівник стабільне і швидке інтернет-з'єднання, чи працює він у зоні з поганим доступом до мережі. Через це процес технічного обслуговування стає складнішим, ніж у випадку пристроїв із фіксованим місцем розташування. Мобільність забезпечує гнучкість для працівників, але одночасно ускладнює технічне обслуговування через зміну місця та мережевих умов (див. рис.1.2, 1.3) [1].

Безпека. У випадку безпеки – кінцеві пристрої, незалежно від того, чи це смартфони, ноутбуки або персональні комп'ютери, часто вразливі до зовнішніх загроз. Основна причина — вони працюють поза захищеною корпоративною мережею, яка зазвичай є закритою і добре захищеною від потенційних атак. Через це пристрої стають основними мішенями для хакерів або інших загроз. Однак для зниження таких ризиків компанії зазвичай впроваджують низку захисних заходів (див. рис.1.8) [8].

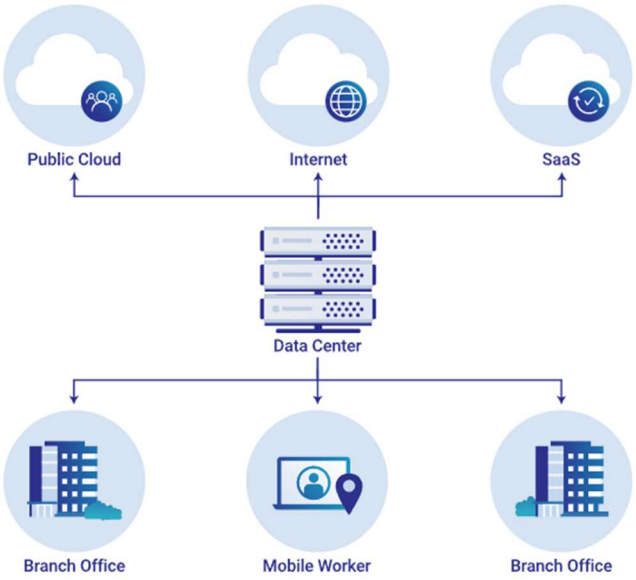

Рисунок 1.8 – Закриті корпоративні мережі та їх ресурси

Зокрема, більшість організацій надають своїм працівникам антивірусне ПЗ. Це може бути стандартне антивірусне рішення, проте часто компанії інвестують у платні корпоративні антивіруси, які включають не тільки базовий захист, але й шифрування даних, фільтрацію трафіку, а також додаткові інструменти для забезпечення безпеки.

Оновлення та підтримка ПЗ. Важливою частиною підтримки безпеки є оновлення системи, встановлення патчів і регулярне технічне обслуговування кінцевих пристроїв. Оновлення безпеки, інсталяція актуального ПЗ, встановлення патчів для усунення вразливостей – все це необхідно для забезпечення стабільної та безпечної роботи кінцевих пристроїв.

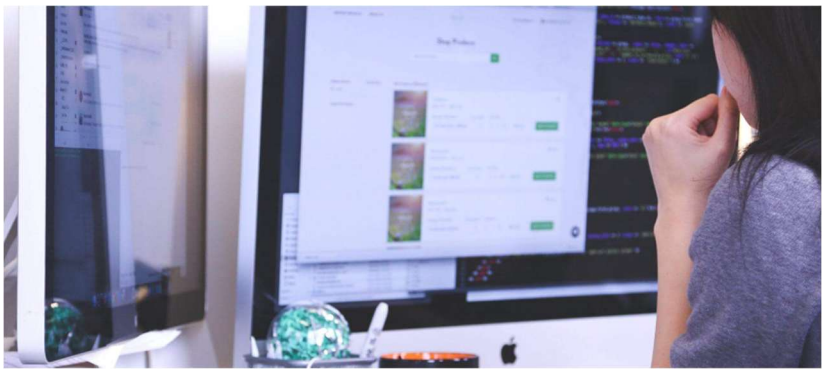

Рисунок 1.9 – Важливість у оновленні та підтримка ПЗ

Антивірусні системи зазвичай працюють у фоновому режимі, виконуючи сканування та оновлення без участі користувача, але такі ньюанси як безпека та патчі, потребують окремого моніторингу і своєчасного втручання з боку технічного персоналу. Безпека є критично важливою, оскільки пристрої працюють у відкритих мережах, і для захисту даних необхідно впроваджувати комплексні рішення, такі як антивірусний захист, шифрування (див. рис.1.9) [9].

Підтримка безпеки кінцевих пристроїв, зокрема оновлення патчів безпеки та баз даних антивірусних програм, має бути постійною. Антивірусні системи повинні регулярно оновлювати свої бази даних, щоб забезпечити захист від нових загроз. Таке завдання зазвичай виконується автоматично, без необхідності безпосереднього втручання технічного персоналу (див. рис.1.9) [9].

Автоматизація процесів оновлення дозволяє значно полегшити технічне обслуговування кінцевих пристроїв, особливо для працівників, які працюють віддалено. Крім антивірусних систем, слід також звернути увагу на регулярні оновлення програмного забезпечення. Оновлення ОС та ПЗ є критично важливим для забезпечення стабільної роботи пристроїв. Якщо програми і системи будуть регулярно оновлюватися, це допоможе уникнути збоїв через застаріле ПЗ або уразливості безпеки. Це особливо актуально для кінцевих пристроїв, які використовуються в умовах віддаленої роботи, де технічне обслуговування не завжди можна виконати фізично [10].

Також слід пам’ятати, що кінцеві пристрої можуть мати різні операційні системи:

— iOS;

— macOS;

— Android;

— Linux;

— Windows.

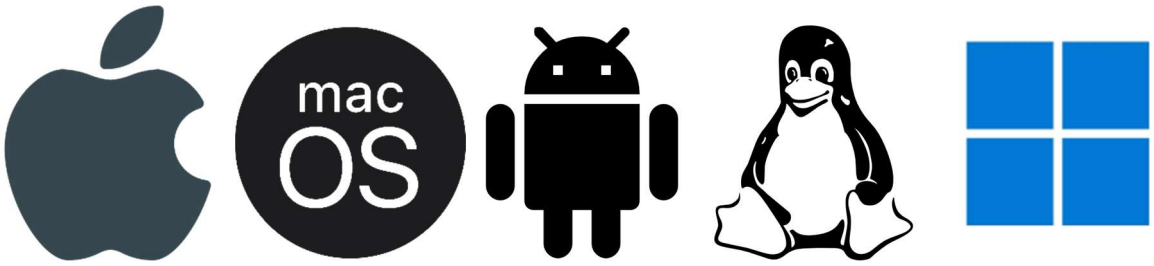

Рисунок 1.10 – Найпопулярніші ОС для кінцевих пристроїв (iOS, macOS, Android, Linux , Windows)

Віддалені працівники можуть використовувати різні пристрої, включаючи ноутбуки та ПК на різних ОС: macOS, Windows та Linux (див. рис.1.4, 1.10) [2, 3]. Найважливішим моментом у цьому є те, щоб працівник адаптував своє робоче середовище до вимог компанії. Наприклад, якщо компанія хоче використовувати лише таку систему, як Linux, співробітникам потрібно буде налаштувати власні пристрої або придбати нові пристрої з відповідною ОС. Якщо ж роботодавець вимагає використання ОС Windows, то працівник зобов’язаний виконувати свої завдання на цій платформі, щоб забезпечити сумісність з корпоративними ресурсами та програмним забезпеченням. Наприклад, можуть виникнути проблеми, якщо працівник, який раніше працював у компанії, що використовувала виключно Linux, переходить на нове місце роботи, де використовується Windows або macOS. Якщо працівники продовжують використовувати Linux, у них можуть виникнути труднощі в роботі з деякими програмами та інструментами, які надає нова компанія. У такому випадку працівники не зможуть ефективно виконувати свої обов'язки, що може негативно позначитися на робочих процесах. Регулярне оновлення системи (див. рис.1.2, 1.3) [1].

Для віддаленої роботи працівник повинен мати відносно сучасний комп'ютер або ноутбук, який здатний виконувати необхідні операції. Не обов'язково володіти надпотужним пристроєм, але важливо, щоб він відповідав мінімальним сучасним системним вимогам, забезпечуючи належну продуктивність для виконання завдань. Звичайно, можна використовувати більш потужний комп'ютер, якщо це потрібно для специфічних завдань, але для більшості робочих процесів буде достатньо пристрою середнього рівня. Оптимальним варіантом може бути ноутбук із наступними параметрами:

- процесор (CPU): Intel Core i5 (10-го покоління або новіше) або AMD Ryzen 5 (4000 серії або новіше).
- оперативна пам'ять (RAM): 8 ГБ – мінімум для базових завдань, 16 ГБ – рекомендується для комфортної роботи.
- накопичувач (SSD): на 256 ГБ або більше.
- графіка (GPU): Вбудована Intel UHD / Iris Xe або AMD Radeon Vega або дискретна NVIDIA GeForce MX450 або найближчі аналоги.
- екран: 14-15.6 дюймів із роздільною здатністю Full HD (1920x1080).
- батарея: час автономної роботи 8-10 годин або більше.

Мобільність кінцевих пристроїв є одночасно і перевагою, і недоліком, особливо з точки зору технічного обслуговування. З одного боку, працівники, які працюють віддалено, отримують свободу переміщень — вони можуть працювати з будь-якої точки світу, використовуючи свої ноутбуки або інші пристрої, що мають автономне живлення, наприклад, акумулятор, і це є безсумнівним плюсом для працівників, адже вони можуть залишатися продуктивними навіть в умовах відсутності доступу до електромережі. Для технічних фахівців, відповідальних за обслуговування цих пристроїв, така мобільність створює певні труднощі (див. рис.1.2, 1.3, 1.4, 1.5) [1, 4].

Оскільки мобільні пристрої часто можуть бути вимкнені або перебувати поза мережею, це ускладнює виконання планового обслуговування, встановлення оновлень чи діагностики. Наприклад, якщо ноутбук буде вимкнено або він втратить заряд, його неможливо буде оновити віддалено. Також можуть виникнути ситуації,

коли пристрій перебуває у зоні з низькою якістю інтернет-з’єднання, що робить процес оновлення або обслуговування ще більш складним. Для цього потрібне вживання додаткових заходів, таких як нагадування працівникам про необхідність підтримувати пристрої в робочому стані або впровадження механізмів, що дозволяють виконувати технічне обслуговування в певні періоди (див. рис.1.2, 1.3, 1.4, 1.5) [1, 4].

Автономність пристроїв, хоч і корисна для користувачів, є певною проблемою для системних адміністраторів. Вони повинні забезпечувати, щоб ці пристрої залишались доступними для виконання критичних операцій навіть тоді, коли працівник не підключений до мережі, або його пристрій тимчасово неактивний. Все це ускладнює процес управління та моніторингу таких пристроїв, оскільки неможливо виконати оновлення або технічне обслуговування у будь-який зручний час (див. рис.1.2, 1.3, 1.4, 1.5) [1, 4].

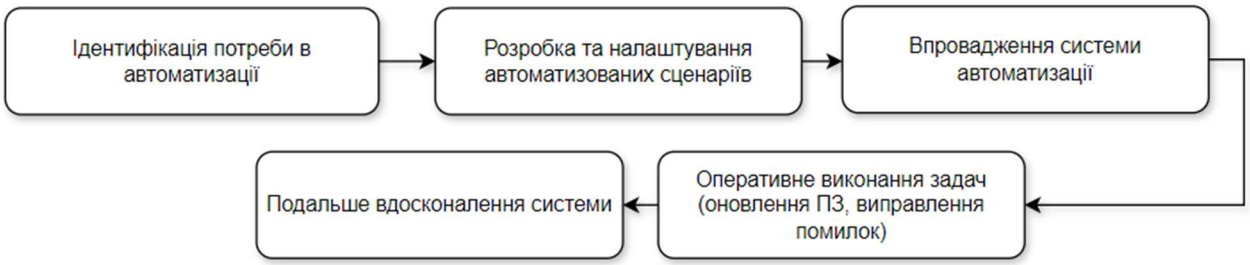

Рисунок 1.11 – Етапи впровадження автоматизації в процес віддаленого обслуговування

1.3 Вимоги та виклики технічного обслуговування кінцевих пристроїв у віддаленому доступі

Коли ми говоримо про вимоги та виклики технічного обслуговування кінцевих пристроїв в умовах віддаленого доступу, стає очевидно, що дистанційна робота і велика кількість віддалених працівників створюють додаткові труднощі. Основні вимоги, що висуваються до технічного обслуговування кінцевих пристроїв, стосуються забезпечення надійності, швидкого реагування на технічні проблеми, безпеки, безперебійності роботи та мінімізації ризиків.

Надійність технічного обслуговування. Для забезпечення стабільної та безперебійної роботи кінцевих пристроїв потрібно своєчасно проводити виявлення та усунення несправностей, профілактичне обслуговування для запобігання небажаним збоям. Надійність також значною мірою залежить від інструментів моніторингу, які надають точні дані про стан системи та надають оперативно вирішувати будь-які недоліки (див. рис. 1.6) [4].

Швидкість реагування на технічні збої. Одна із ключових вимог в розподілених системах, особливо коли кінцеві пристрої знаходяться в різних локаціях і можуть постійно змінювати своє місце розташування, що вимагає оперативного втручання для мінімізації часу простою та відновлення роботи. Працівники повинні мати змогу продовжувати свою діяльність без перерв, навіть у разі виникнення технічних проблем. Для цього необхідно запровадити механізми автоматизованого моніторингу, швидкого виявлення несправностей та віддаленої діагностики, щоб забезпечити оперативне вирішення технічних питань (див. рис. 1.4, 1.6) [1, 4].

Безпека. Захист даних і систем на кінцевих пристроях особливо важливий при віддаленій роботі. Це включає регулярне оновлення ПЗ, захист від шкідливих програм, шифрування даних і додаткові заходи безпеки, такі як двофакторна автентифікація і VPN для безпечного доступу до корпоративної мережі (див. рис.1.9) [8]. Технічне обслуговування повинно передбачати регулярне оновлення

патчів безпеки, перевірку наявності вірусів та загроз, а також контроль за доступом до пристроїв і даних, що на них зберігаються (див. рис. 1.6) [4, 8].

Безперебійність роботи. Одне із важливим завданням технічного обслуговування, особливо для корпоративних рішень та критично важливих бізнес-процесів, які не можуть допускати простоїв. Для цього необхідно запровадити систему моніторингу, профілактичні заходи та регулярне резервне копіювання даних. Такі заходи дозволяють швидко відновити роботу після збоїв без ризику втрати або витоку даних (див. рис. 1.6) [4, 8].

Мінімізація ризиків. Щоб забезпечити безперебійну роботу, необхідно врахувати не лише загальні питання безперебійності, а й мінімізацію ризиків, пов'язаних із технічними збоями та втратою даних, що особливо важливо у випадках втрати мережевого доступу або несправності обладнання. Як було зазначено раніше, одним із ключових рішень для таких ситуацій є регулярне резервне копіювання даних. Створення копій системи дозволяє швидко відновити її на будь-якому іншому пристрої за потреби, забезпечуючи стабільність та надійність роботи (див. рис. 1.4, 1.6) [1, 4, 8].

Простота та доступність. Ще однією важливою вимогою до технічного обслуговування є його простота та доступність. Обслуговування кінцевих пристроїв має бути максимально зручним для користувачів, щоб зменшити їхню участь у процесах вирішення технічних проблем. Наприклад, для автоматизації усунення недоліків або встановлення додаткового програмного забезпечення можна використовувати спеціальні сценарії чи скрипти. Вони можуть виконуватися у фоновому режимі, без необхідності взаємодії користувача з інтерфейсами чи процесами. Тобто, всі необхідні дії будуть виконуватись автоматично, а працівники навіть не помічатимуть цих процесів, якщо обслуговуючий персонал не залишить відповідних повідомлень чи коментарів у системі.

Сумісність з будь-якими типами пристроїв. Важливо зрозуміти, що інколи вимоги до роботи з конкретними операційними системами можуть диктуватися самою компанією або роботодавцем. Наприклад, компанія може вимагати

використовувати лише операційні системи на базі Windows, macOS або дистрибутивів Linux. Однак, є ситуації, коли необхідно забезпечити підтримку різних типів кінцевих пристроїв, і це стосується не тільки операційних систем, а й самих пристроїв, таких як ноутбуки, персональні комп'ютери, стаціонарні та мобільні пристрої тощо. У таких випадках потрібно забезпечити технічне обслуговування, яке б підтримувало мультиплатформеність. Система технічної підтримки повинна ефективно працювати з різними операційними системами, як зазначалося раніше, а також з різноманітними апаратними конфігураціями незалежно від рівня їхньої складності. Така універсальність дозволяє охопити широкий спектр пристроїв, забезпечуючи їхню ефективну роботу та мінімізацію можливих збоїв чи конфліктів між програмним забезпеченням і апаратними компонентами (див. рис. 1.3, 1.5, 1.6, 1.10) [2, 3, 4, 9, 10].

Для проведення якісної оцінки технічного обслуговування важливо глибше аналізувати загальноприйняті стандарти та вимоги, які використовуються в цій сфері.

Оцінка якості обслуговування може включати такі ключові параметри [11]:

1. Середній час ремонту (MTTR) – це середній час, необхідний для вирішення технічної проблеми і повернення обладнання до нормальної роботи. Чим коротший цей час, тим ефективніше обслуговування. Швидке відновлення зводить до мінімуму час простою і підтримує роботу системи.

2. Середній час напрацювання на відмову (MTBF) – це показник середнього інтервалу між технічними відмовами в роботі обладнання: Вище значення MTBF свідчить про вищу надійність в експлуатації, а довший MTBF означає, що обладнання є більш стабільним. З іншого боку, низький показник MTBF вказує на те, що збої трапляються частіше і вимагають кращого обслуговування.

3. Кількість інцидентів безпеки – це безпосередньо стосується кількості виявлених і усунутих загроз для безпеки. Зменшення кількості таких інцидентів є показником того, що профілактичні заходи та система моніторингу працюють ефективно, що безпосередньо впливає на якість обслуговування та захист даних.

4. Час простою (downtime) – показник, який відображає тривалість простою системи або пристрою через технічні збої. Чим коротший час простою, тим вища продуктивність системи та задоволеність користувачів. Навпаки, довший час простою свідчить про низьку ефективність обслуговування і негативно впливає на користувачів.

5. Частота оновлення – це параметр, який вимірює швидкість і регулярність оновлення програмного забезпечення, включаючи патчі безпеки та інші важливі компоненти. Регулярні та швидкі оновлення системи важливі для забезпечення стабільної та безпечної роботи пристрою і є важливим аспектом обслуговування.

1.4 Роль технологій автоматизації в управлінні розподіленими системами

Вплив технологій автоматизації на управління розподіленими системами є надзвичайно значущим, оскільки такі системи складаються з багатьох кінцевих пристроїв, розташованих у різних фізичних локаціях і працюючих на різних платформах. У цих умовах ручне управління та технічне обслуговування стають вкрай складними, ресурсоємними та схильними до людських помилок, що може призвести до непередбачуваних збоїв. Технології автоматизації дозволяють спростити та оптимізувати ці процеси за рахунок сучасних підходів і інструментів.

Централізоване управління. Однією з ключових переваг автоматизації є можливість централізованого управління – використання систем централізованого управління дає змогу контролювати стан усіх кінцевих пристроїв у реальному часі, здійснювати моніторинг продуктивності, перевіряти оновлення безпеки та оперативно реагувати на інциденти з мінімальними витратами [4, 8, 9 10, 12]. Централізовані рішення також дозволяють автоматизувати конфігурацію будь-

яких пристроїв, значно скорочуючи час і зусилля, необхідні для індивідуального налаштування кожного елемента системи.

Моніторинг Автоматизований моніторинг є важливим компонентом ефективного управління розподіленими системами. Сучасні системи моніторингу можуть забезпечити раннє виявлення потенційних проблем і запобігти більш серйозним збоям, відстежуючи різні параметри, такі як завантаження процесора, використання пам’яті, підключення до мережі, доступність пристроїв, стан дисків та інші показники. Крім того, автоматизовані системи моніторингу можуть включати механізми, які прогнозують потенційні збої на основі зібраних даних, забезпечують профілактичне обслуговування, зменшують ймовірність перебоїв і підвищують стабільність системи. [4, 12].

Оновлення та підтримка ПЗ в актуальному стані. Автоматизація також значно спрощує обслуговування кінцевих точок. Наприклад, регулярні оновлення ОС, ПЗ та патчів безпеки можуть здійснюватися централізовано і автоматично [4, 9, 10, 12].

Безпека. Коли йдеться про безпеку розподілених систем, автоматизовані рішення відіграють ключову роль у забезпеченні їх захисту – завдяки автоматизованим засобам можна здійснювати постійний моніторинг загроз, виконувати регулярні оновлення безпеки та контролювати доступ до мережевих ресурсів. Виявлення загроз і реагування на них можуть бути повністю автоматизованими, що дозволяє швидко виявляти шкідливі дії та негайно вживати заходів для їх нейтралізації [4, 8, 12].

Зменшення економічних витрат. Автоматизація технічного обслуговування дозволяє значно заощадити час як для працівників, так і для компанії – автоматизація допомагає скоротити фінансові витрати, які можуть бути перенаправлені на інші потреби компанії або використані для підвищення зарплат працівникам. Тому впровадження автоматизованих рішень у технічне обслуговування є вигідним для всіх сторін: як для працівників, так і для роботодавців, що сприяє підвищенню загальної ефективності компанії [12].

РОЗДІЛ 2 МЕТОДИ ТА ЗАСОБИ АВТОМАТИЗАЦІЇ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ КІНЦЕВИХ ПРИСТРОЇВ

2.1 Огляд сучасних методів автоматизації технічного обслуговування

Огляд існуючих методів автоматизації обслуговування кінцевих пристроїв, які використовують віддалені працівники, у розподіленому робочому середовищі підкреслює важливість використання інструментів, які зменшують ручні зусилля та забезпечують надійність пристрою. Ефективна автоматизація в цій сфері передбачає розгортання централізованих систем керування конфігурацією, автоматизацію оновлень програмного забезпечення та постійний моніторинг стану пристрою [13].

Розгортання стандартних конфігурацій. Централізоване керування конфігурацією має важливе значення для узгодженої та безпечної роботи кінцевих пристроїв, які використовуються віддаленими працівниками. Microsoft Endpoint Manager (включає Intune (у хмарі) і Microsoft Endpoint Configuration Manager (локально), або MECM) — це ультимативне рішення, яке забезпечує централізоване керування розгалуженими мережами пристроїв. За допомогою Endpoint Manager IT-команди можуть віддалено розгортати стандартні конфігурації на таких пристроях, як ноутбуки, ПК та мобільні пристрої (див. рис. 2.1) [13, 14].

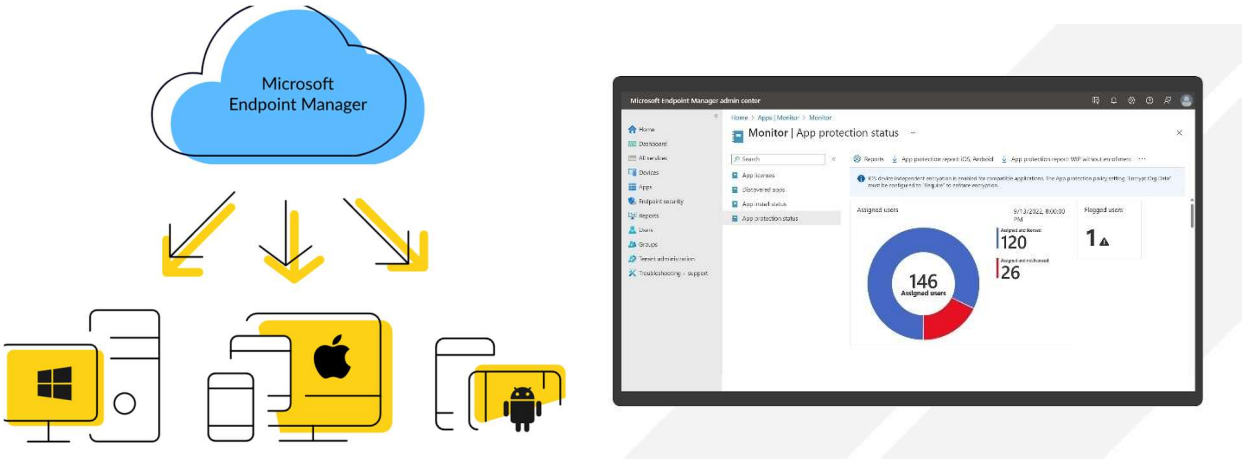

Рисунок 2.1 – Автоматизоване технічне обслуговування кінцевих пристроїв за допомогою Microsoft Endpoint Manager (Intune + MECM)

Наприклад, коли новий ноутбук надається новому віддаленому працівнику, він надходить із попередньо налаштованими основними політиками безпеки та необхідним програмним забезпеченням. За допомогою Endpoint Manager IT-команди можуть забезпечити дотримання корпоративних стандартів, таких як права доступу, конфігурації брандмауера, налаштування VPN та інші важливі заходи безпеки, без безпосередньої участі користувача – це централізоване налаштування забезпечує безпечні, сумісні пристрої з мінімальними перешкодами для робочого процесу працівників (див. рис. 1.7, 2.1) [8, 13, 14].

Автоматизація розгортання ПЗ. Автоматичне встановлення програмного забезпечення є ще однією важливою функцією віддаленого керування пристроями. За допомогою Microsoft Intune IT-команди можуть віддалено розгортати необхідні програми на пристроях працівників, незалежно від того, чи є вони на ноутбуках, ПК чи мобільних пристроях. Така функціональність є безцінною для територіально розосередженої робочої сили, де ручне розгортання програмного забезпечення є недоцільним. Наприклад, якщо компанії потрібно представити нове ПЗ або оновити існуючі програми, IT-команда може автоматично поширювати ці оновлення на всі відповідні пристрої, оптимізуючи підтримку та мінімізуючи збої (див. рис. 2.1) [14].

Управління конфігураціями через політики. Застосовуючи політики через Microsoft Endpoint Configuration Manager (MECM), IT-команди можуть

застосовувати конфігурації пристроїв, щойно працівники підключаться до корпоративної мережі чи Інтернету. Такий підхід особливо важливий у контексті віддаленої роботи, де він гарантує, що всі пристрої залишаються у відповідності до стандартів і налаштувань, обраними компанією, незалежно від місця розташування їхніх працівників (див. рис. 1.7, 2.1) [8, 14].

Автоматизація процесу оновлень. На віддалених пристроях є критично важливо забезпечити автоматизацію оновлень для забезпечення безпеки та стабільної роботи корпоративної інфраструктури. Віддалені пристрої постійно потребують оновлень для виправлення вразливостей, покращення продуктивності та додавання нових функцій, і системи автоматизації дозволяють встановлювати ці оновлення без втручання користувачів [14].

Для оновлень програмного забезпечення MECM забезпечує автоматичне розповсюдження оновлень для основних програм, наприклад антивірусного програмного забезпечення. Автоматизація оновлень гарантує, що всі віддалені пристрої працюють з останніми версіями програмного забезпечення, зменшуючи ризик, пов’язаний із застарілими програмами, і підвищуючи безпеку в організації, де це застосовується.

Віддалений моніторинг стану здоров’я. Моніторинг справності віддалених пристроїв має вирішальне значення для виявлення та вирішення проблем до того, як вони вплинуть на продуктивність працівників. Microsoft Endpoint Manager включає можливості моніторингу; однак існують спеціалізовані сторонні інструменти, такі як Splunk, Zabbix і Nagios (див. рис. 2.2), які можуть бути інтегровані для надання додаткової інформації. Такі інструменти пропонують широкі можливості моніторингу, такі як відстеження навантаження ЦП, використання пам’яті та активності диска, що допомагає виявити аномалії продуктивності [15, 16].

Рисунок 2.2 – ПЗ для моніторингу
(Splunk, Zabbix і Nagios)

Варто уточнити, що Splunk, Zabbix і Nagios (див. рис. 2.2) є додатковими рішеннями для моніторингу і не замінюють основні функції керування пристроями Microsoft Endpoint Manager. Натомість вони забезпечують розширений моніторинг системних показників і журналів, що може розширити можливості Endpoint Manager [15, 16].

Моніторинг використання ресурсів. Моніторинг використання ресурсів дозволяє IT-командам виявляти проблеми з продуктивністю на віддалених ноутбуках і ПК. Відстежуючи ключові показники, такі як завантаження процесора, використання оперативної пам'яті та активність диска, такі інструменти, як Splunk, Zabbix і Nagios, можуть попередити IT про високе використання ресурсів або незвичайні шаблони, як часті перезавантаження. Такі сповіщення дозволяють IT-службам завчасно вирішувати проблеми, забезпечуючи мінімізацію ризиків роботи кінцевих пристроїв, які використовують віддалені працівники [15, 16].

Моніторинг стану мережі та підключень. Моніторинг мережевої активності кінцевих пристроїв є важливим для підтримки безпечного та надійного з'єднання. Такі інструменти, як Zabbix, можуть відігравати важливу роль, відстежуючи VPN-з'єднання, забезпечуючи підключення пристроїв через захищені канали. Цей тип моніторингу особливо важливий у віддаленому робочому середовищі, де безпечний і послідовний доступ до корпоративних ресурсів є пріоритетом. Якщо пристрій втрачає з'єднання з мережею, Zabbix може надсилати автоматичні

сповіщення до IT-командів, дозволяючи своєчасно втручатися та мінімізувати можливі простої [8, 15, 16].

Моніторинг на рівні системних журналів. Такі рішення, як Splunk, забезпечують надійний аналіз журналів, постійно скануючи системні журнали подій на наявність аномалій, помилок або підозрілої активності. Автоматизуючи аналіз журналів, Splunk може попереджати IT-команди про потенційні проблеми, такі як ризики безпеки чи збої в роботі системи, не вимагаючи від них вручну перевіряти журнали на кожному пристрої. Така можливість є цінною для віддалених пристроїв, оскільки дозволяє IT-командам завчасно виявляти та вирішувати проблеми [16].

Ключові функції систем віддаленого моніторингу:

— контроль стану жорстких дисків;

— попередження про можливі збої;

— автоматичне створення завдань для резервного копіювання даних.

Моніторинг стану жорсткого диска. Постійне відстеження стану жорсткого диска, що допомагає запобігти втраті даних і підтримує профілактичне обслуговування.

Попередження про збій. Рішення для моніторингу, яке може виявляти ранні ознаки збою системи та надсилати сповіщення, дозволяючи IT-командам вживати запобіжних заходів.

Автоматичне створення завдань резервного копіювання даних. Рішення для моніторингу, що може автоматично ініціювати завдання резервного копіювання для забезпечення цілісності даних.

Підтримка мобільних пристроїв. Окрім традиційних кінцевих пристроїв, таких як ноутбуки та ПК, керування мобільними пристроями (Mobile Device Management, MDM) має вирішальне значення в контексті віддаленої роботи. Microsoft Intune надає комплексні функції MDM, керуючи мобільними пристроями на платформах iOS і Android (див. рис. 2.3) [2, 3, 17]. Intune підтримує застосування політики, налаштування безпечного доступу, віддалене стирання та розгортання

програм для мобільних пристроїв, гарантуючи, що вони відповідають політикам компанії, як і традиційні кінцеві пристрої – ноутбуки та ПК (див. рис. 2.1) [14].

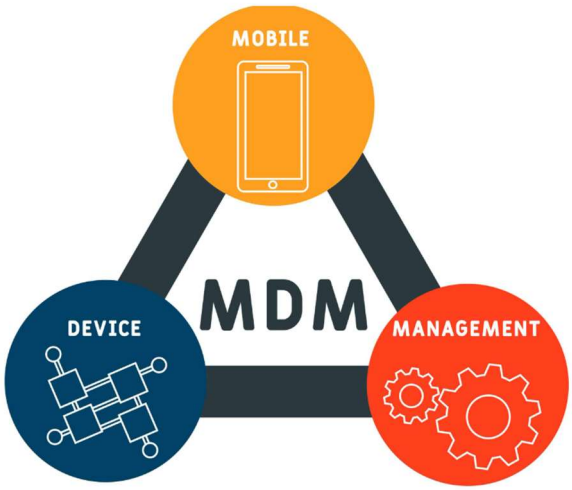

Рисунок 2.3 – ПЗ для керування мобільними пристроями (Mobile Device Management (MDM))

Автоматизація обслуговування кінцевих пристроїв, які використовують віддаленні працівники, дозволяє IT-командам ефективно керувати й підтримувати глобально розподілену систему віддалених працівників. Microsoft Endpoint Manager (Intune + MECM) надає основні функції для централізованої конфігурації, розгортання програмного забезпечення, автоматизації оновлень і керування мобільними пристроями, тоді як такі інструменти, як Splunk, Zabbix і Nagios, служать спеціалізованими рішеннями для моніторингу, пропонуючи покращене «бачення» продуктивності пристрою та стану мережі. Разом ці інструменти забезпечують надійний, спрощений підхід до підтримки безпечного та ефективного віддаленого IT-середовища [14, 15, 16, 17].

2.2 Використання хмарних технологій для обслуговування кінцевих пристроїв

Використання хмарних технологій для обслуговування кінцевих пристроїв значно спрощує процеси технічного обслуговування, особливо в умовах віддаленої роботи. Таке рішення дозволить централізовано управляти пристроями незалежно від їх типу та фізичного розташування працівників, забезпечуючи віддалене резервне копіювання, оновлення систем і безпеку пристроїв. Такий підхід є особливо зручним, оскільки працівники можуть використовувати різноманітні пристрої, такі як ноутбуки, персональні комп'ютери, смартфони та інші, які потребують постійного контролю і моніторингу для забезпечення їхньої безперебійної роботи.

У компаніях, де працівники працюють віддалено і не мають можливості приїжджати до офісу для налаштування або ремонту пристроїв, хмарні технології дозволяють надавати готові до використання пристрої з уже попередньо встановленими налаштуваннями, що мінімізує необхідність фізичної взаємодії з технічним персоналом і значно підвищує ефективність обслуговування, адже більшість технічних процесів, таких як встановлення оновлень або налаштування безпеки, виконуються автоматизовано.

Ключовим аспектом цієї автоматизації є використання хмарних рішень для централізованого управління пристроями. Наприклад, такі платформи, як Microsoft Intune або Google Workspace (Google Endpoint Management), забезпечують можливість централізованого керування робочими пристроями працівників – вони дозволяють IT-відділу контролювати оновлення систем, встановлювати політики безпеки, управляти доступом до корпоративних даних і багато іншого, незалежно від місця знаходження працівників (див. рис. 2.4) [14, 17, 19, 20].

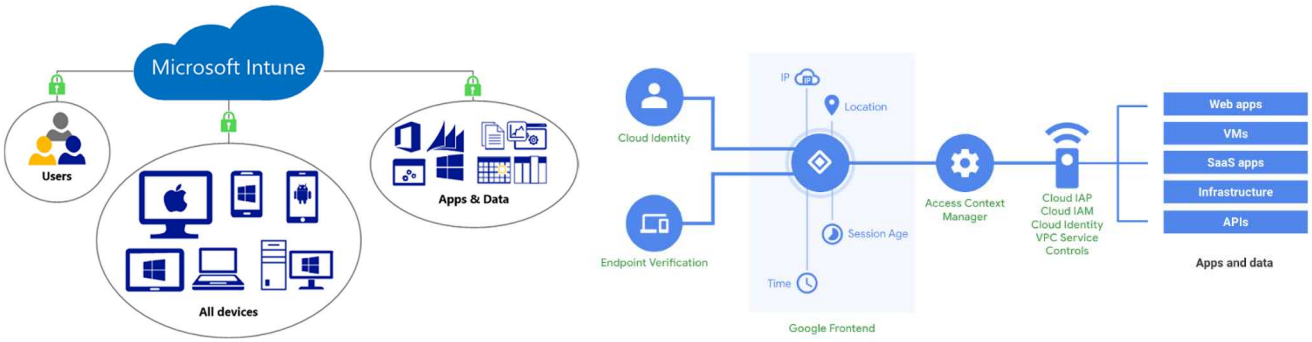

Рисунок 2.4 – Хмарні рішення по адмініструванню кінцевих пристроїв (Mobile Device Management (MDM) та Google Endpoint Management)

Окрім керування пристроями, хмарні технології також надають інструменти для резервного копіювання та захисту даних. Рішення забезпечує автоматичне резервне копіювання та зберігання важливих файлів у хмарному сховищі, що захищає від втрати даних у разі пошкодження або виходу пристрою з ладу. Хмарне рішення ефективно підтримує високий рівень безпеки та стабільність бізнес-процесів навіть при віддаленій роботі [14, 17, 18, 19].

Якщо говорити більш детально про платформи для віддаленого управління пристроями, такі як Microsoft Intune і Google Workspace (Google Endpoint Management) (див. рис. 2.4), однією з їхніх ключових переваг є можливість централізованого керування всіма кінцевими пристроями організації через хмару. Ці платформи надають IT-відділу інструменти для забезпечення безпеки та стабільної роботи пристроїв, включаючи оновлення, застосування політик безпеки та моніторинг стану пристроїв у реальному часі [18, 19].

Microsoft Intune. Платформа забезпечує централізоване управління пристроями через єдиний портал, що дозволяє IT-командам керувати пристроями, які використовують різні ОС, з єдиної інформаційної панелі. Наприклад, віддалені працівники, які використовують ноутбуки, можуть автоматично отримувати оновлення конфігурації та політики безпеки. Адміністратори можуть віддалено застосовувати необхідні налаштування, активувати політики безпеки і навіть виконувати важливі дії для захисту конфіденційної інформації, наприклад, стирати дані з втрачених або викрадених пристроїв [14, 17, 18].

Що стосується безпеки, інтеграція Microsoft Intune з Entra ID (раніше Azure AD) дозволяє використовувати політики умовного доступу, тобто система автоматично перевіряє відповідність пристрою певним вимогам безпеки, таким як наявність антивірусного захисту, шифрування дисків або встановлення останніх оновлень. Наприклад, якщо ноутбук працівника не відповідає встановленим стандартам безпеки, доступ до корпоративних ресурсів може бути тимчасово обмежений до тих пір, поки пристрій не пройде необхідні перевірки та оновлення, що гарантуватиме, що всі віддалені пристрої функціонують відповідно до високих стандартів безпеки компанії (див. рис. 2.4, 2.5) [14, 17, 18, 19].

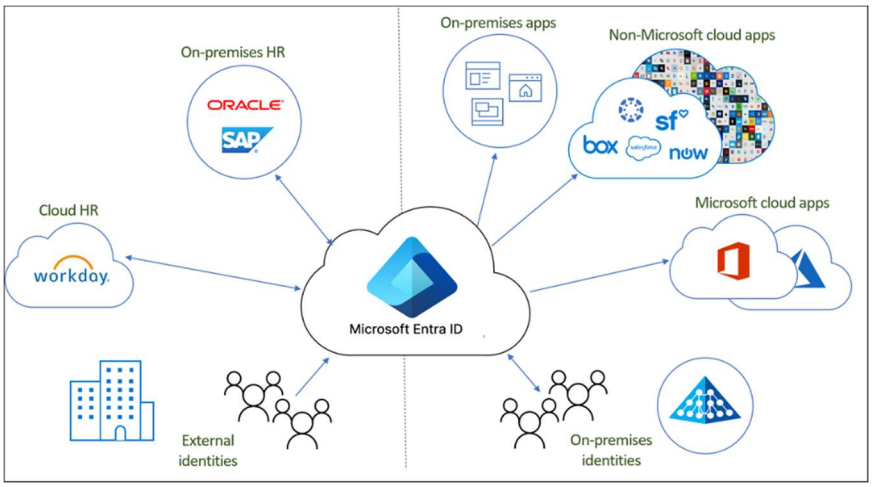

Рисунок 2.5 – Хмарний сервіс керування ідентифікацією та доступом (Microsoft Entra ID)

Microsoft Intune також спрощує процес оновлення та розгортання програмного забезпечення та IT-відділ може автоматично розгортати нові додатки або оновлювати існуючі на віддалених пристроях без необхідності взаємодії з кожним користувачем окремо. Дане рішення особливо корисно в умовах, коли велика кількість віддалених працівників потребує своєчасних оновлень критично важливих інструментів для роботи. Завдяки таким можливостям, компанії можуть підтримувати актуальність програмного забезпечення на всіх пристроях, зменшуючи ризик помилок або вразливостей через застарілі версії програм (див. рис. 2.4) [14, 18].

Google Workspace (Google Endpoint Management). Ця платформа надає потужні інструменти для управління мобільними пристроями (MDM). Вона дозволяє адміністраторам контролювати доступ до корпоративних ресурсів на мобільних пристроях, включаючи платформи Android та iOS. Адміністратори можуть накладати обмеження на використання певних додатків і забезпечувати дотримання корпоративних політик безпеки. У разі загрози або втрати пристрою адміністратори можуть віддалено стерти конфіденційні дані, щоб запобігти витоку (див. рис. 2.4) [17, 19].

Щодо забезпечення безпеки даних, Google Workspace (Google Endpoint Management) пропонує інструменти для резервного копіювання та синхронізації інформації працівників, що гарантує її збереження та доступність незалежно від того, який пристрій використовується і де саме знаходиться працівник, що особливо корисно в умовах віддаленої роботи, оскільки працівники можуть бути в будь-якому куточку світу, але все одно мати надійний доступ до корпоративних даних. У разі втрати або пошкодження пристрою, всі дані можна легко відновити через Google Drive, що забезпечує безперервність робочого процесу і мінімізує втрати інформації.

Хмарні рішення для резервного копіювання, такі як Amazon S3, Azure Backup та Google Drive, відіграють ключову роль у забезпеченні безпеки та доступності даних працівників, особливо в умовах віддаленої роботи. Однією з основних вимог до таких рішень є гарантування збереження даних і можливість їх швидкого відновлення після збою. Завдяки хмарним рішенням процес резервного копіювання можна легко автоматизувати, що робить збереження даних на віддалених пристроях більш надійним і гнучким [21].

Рисунок 2.6 – Хмарні сервіси резервування даних
(Amazon S3, Azure Backup, Google Drive)

Amazon S3. Це рішення дозволяє зберігати критичні файли кожного працівника в хмарі. Працівники можуть налаштовувати автоматичне резервне копіювання важливих робочих документів на Amazon S3, забезпечуючи їх збереження навіть у випадках збою або втрати локального пристрою. До того ж, Amazon S3 підтримує версіонування файлів, що дозволяє зберігати кілька версій одного й того ж документа, що особливо корисно у випадках випадкового видалення чи пошкодження файлів, адже завжди можна відновити попередню версію документа (див. рис. 2.6) [21].

Azure Backup. Ще одне потужне рішення для резервного копіювання, яке інтегрується з Microsoft 365 і дозволяє автоматично створювати резервні копії даних на віддалених пристроях. Зокрема, пристрої на базі Windows можна налаштувати на регулярне резервне копіювання важливих бізнес-файлів, системних налаштувань і баз даних до Azure Backup, забезпечуючи централізоване зберігання даних і швидке відновлення в разі збою. Крім того, інтеграція з Microsoft 365 дозволяє створювати резервні копії поштових скриньок, файлів OneDrive і SharePoint, зберігаючи всю важливу інформацію в одному місці (див. рис. 2.6) [21].

Google Drive. Найбільш поширеним хмарним рішенням для резервного копіювання є Google Drive, з яким знайомі більшість користувачів. Google Drive дозволяє автоматично синхронізувати дані працівників з хмарою, і це надзвичайно зручне рішення для віддалених працівників, оскільки вони можуть зберігати свої документи у хмарі та мати до них доступ з будь-якого пристрою. Він також підтримує функцію спільної роботи над файлами, що дозволяє кільком

працівникам одночасно працювати над одним і тим же документом. Файли можуть бути як розміщені безпосередньо у Google Drive, так і автоматично синхронізовані з локальними пристроями, що робить цей процес зручним і ефективним (див. рис. 2.6).

Подібно до Google Drive, Microsoft OneDrive, що інтегрований з підпискою Microsoft 365, також надає можливість автоматичної синхронізації та збереження даних. Користувачі OneDrive можуть отримати до 1 Тб місця для зберігання, що дозволяє синхронізувати важливі файли і папки з OneDrive та зберігати їх у хмарному сховищі автоматично, що робить таке рішення особливо зручним для тих, хто використовує екосистему Microsoft, забезпечуючи безпечне збереження даних та їх легкий доступ з будь-якого пристрою [18, 20].

Хмарні рішення для кібербезпеки, такі як CloudFlare та Cisco Umbrella, відіграють важливу роль у захисті кінцевих пристроїв, які використовуються віддаленими працівниками, що стає особливо критичним у сучасних умовах, коли працівники можуть знаходитися у різних місцях, а їхні пристрої повинні мати захищений доступ до корпоративних ресурсів. Завдяки цим хмарним рішенням [44], компанії можуть забезпечити безпеку мережі та захист від кіберзагроз незалежно від місця розташування працівників (див. рис. 2.7) [8, 22].

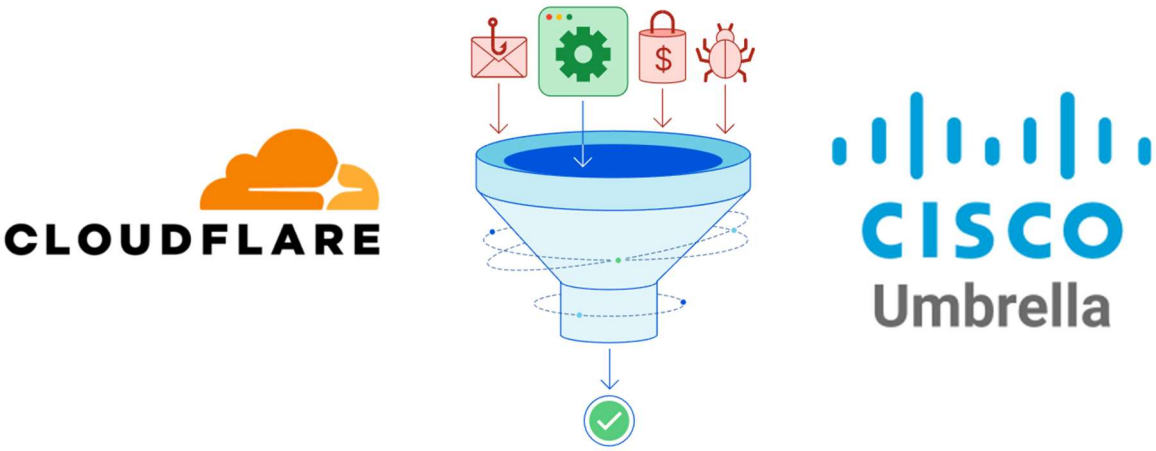

Рисунок 2.7 – Хмарні сервіси для захисту кінцевих пристроїв (CloudFlare, Cisco Umbrella)

CloudFlare. Це хмарне рішення для кібербезпеки відоме завдяки своєю здатністю забезпечувати надійний захист мережевих з'єднань та веб-додатків. Захист мережевих з'єднань від таких загроз, як DDoS-атаки, фішинг і інші види кіберзлочинності, досягається за допомогою швидкого шифрування мережевого трафіку, що особливо корисно для віддалених працівників, які підключаються до корпоративних ресурсів через Інтернет. Завдяки Cloudflare, вони можуть безпечно працювати навіть за межами офісу, а їхня інформація залишається захищеною. Cloudflare надає захист веб-додатків, що використовуються віддаленими працівниками, гарантуючи безпечний доступ до корпоративних систем і захищаючи їх від атак на веб-сайти та додатки [8, 22].

Cisco Umbrella. Також пропонує важливі функції для кібербезпеки. Однією з ключових переваг є можливість виявлення загроз у реальному часі та захист на рівні DNS, тобто Umbrella може проактивно виявляти загрози, аналізуючи мережеву активність кінцевих пристроїв. Наприклад, якщо віддалений працівник намагається підключитися до шкідливого веб-ресурсу, система блокує цей доступ і попереджає користувача про небезпеку, що схоже на те, як Google попереджає користувачів про небезпечні або фішингові сайти, однак Umbrella діє на ще більш ранньому етапі, аналізуючи DNS-запити до того, як працівник відвідає небезпечний сайт. Завдяки цьому, віддалені працівники мають додатковий рівень захисту ще до того, як їхні пристрої можуть потрапити під загрозу [22].

2.3 Програмні та апаратні інструменти моніторингу та управління

Хмарні технології є надзвичайно зручними для використання, особливо в контексті віддаленої роботи, але не варто забувати про певні обмеження. Наприклад, ситуації, коли можна втратити доступ до мережі, що унеможливлює використання хмарних сервісів. До того ж, деякі налаштування доводиться здійснювати безпосередньо на рівні програмного забезпечення, яке встановлене на

кінцевих пристроях, або навіть на апаратних рішеннях, вбудованих на етапі виробництва.

Часто виробники забезпечують пристрої власними інструментами для моніторингу і управління, що потребує додаткової уваги. У процесі технічного обслуговування кінцевих пристроїв, які використовуються віддаленими працівниками, незалежно від того, чи використовують вони пристрої, надані компанією, чи власне обладнання, важливим є інтеграція як програмних, так і апаратних рішень для моніторингу та управління. Такі інструменти дозволяють не лише спостерігати за станом пристроїв, але й забезпечувати їхню безпеку, своєчасно виконувати оновлення, а також діагностувати потенційні проблеми в режимі реального часу. Завдяки цим рішенням можна ефективно забезпечувати безперервну роботу віддалених пристроїв та мінімізувати ризики виникнення технічних несправностей, виконуючи всі необхідні дії у віддаленому режимі.

Якщо розглядати сучасні платформи для управління пристроями, а їх чимало, варто звернути увагу на такі рішення:

— Google Workspace (Endpoint Management) ;
— Microsoft Endpoint Manager;
— VMware Workspace ONE;
— Cisco Meraki Systems Manager;
— SolarWinds N-central;
— ManageEngine Endpoint Central;
— Jamf Pro;
— Ivanti Unified Endpoint Manager;
— Citrix Endpoint Management;
— Sophos Mobile;
— BlackBerry UEM;
— Hexnode UEM;
— Baramundi Management Suite;
— SOTI MobiControl;

— Kaseya VSA;

— Trusted Platform Module (TPM).

Google Workspace (Endpoint Management). Google Workspace (управління кінцевими точками) надає базові функції управління мобільними пристроями (MDM) для пристроїв Chrome, Android, iOS і Windows. Адміністратори можуть керувати програмами, впроваджувати політики безпеки та контролювати їх дотримання через Google Cloud. Він не підтримує комплексне дистанційне керування, але базові функції, такі як блокування та очищення загублених пристроїв, є можливими.

Автоматичні оновлення додатку Google Workspace забезпечують безпеку всіх пристроїв користувачів, що робить його хорошим варіантом для організацій, які вкладають значні кошти в сервіси Google і Chrome OS (див. рис. 2.8) [23].

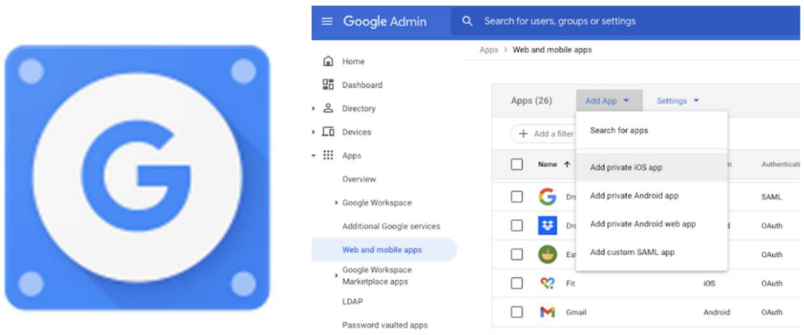

Рисунок 2.8 – Google Workspace (Endpoint Management) (логотип та інтерфейс)

Microsoft Endpoint Manager. Інтегроване рішення, яке поєднує Microsoft Intune (Mobile Device Management) з Configuration Manager для традиційного керування робочим столом, що робить його придатним для керування різноманітними пристроями, такими як ПК з Windows, Mac, iOS та Android. Endpoint Manager автоматизує розгортання програмного забезпечення, оновлення та впроваджує політики безпеки як на керованих, так і на некерованих пристроях. Його комплексні функції розгортання додатків і забезпечення безпеки дозволяють IT-командам налаштовувати засоби контролю доступу, встановлювати політики умовного доступу та керувати відповідністю для віддалених працівників і

пристроїв корпоративного офісу. Крім того, він надає засоби дистанційного усунення несправностей, що дозволяє виконувати такі дії, як перезавантаження, інсталяція програмного забезпечення або видалення програм за потреби (див. рис. 2.9) [23].

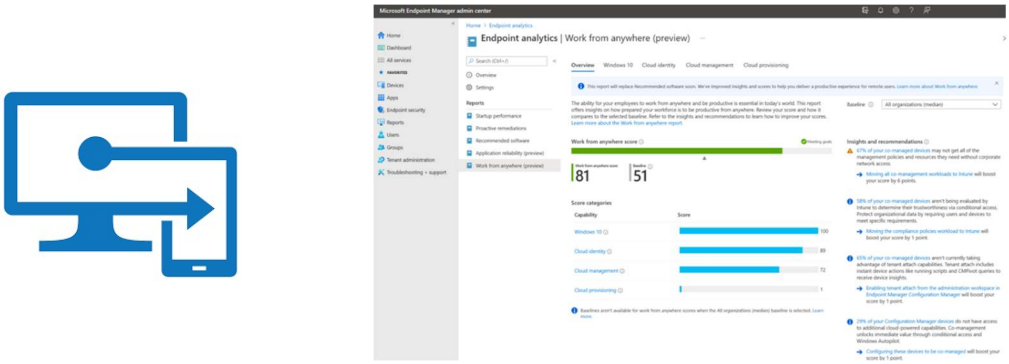

Рисунок 2.9 – Microsoft Endpoint Manager (логотип та інтерфейс)

До того ж, політики умовного доступу Endpoint Manager захищають конфіденційні дані, вимагаючи відповідності пристрою перед доступом до корпоративних ресурсів. Управління виправленнями повністю автоматизоване, забезпечує регулярне оновлення ОС і додатків і закриває вразливості. Ця функціональність забезпечує безпеку для віддалених пристроїв, навіть якщо вони знаходяться за межами корпоративної мережі, зберігаючи безперервність дотримання організаційних стандартів безпеки.

VMware Workspace ONE. Уніфікована платформа для управління кінцевими точками (UEM), призначена для безперешкодного управління різноманітними пристроями, включаючи ПК, ноутбуки, мобільні пристрої та пристрої Інтернету речей. Тісна інтеграція з Windows, macOS, iOS, віртуальною та хмарною інфраструктурою VMware дозволяє IT-командам віддалено виконувати такі дії, як встановлення програм, перезавантаження та налаштування параметрів пристроїв. Конфігурацію та інші дії можна виконувати віддалено, щоб забезпечити відповідність нормативним вимогам і безпеку. (див. рис. 2.10) [23].

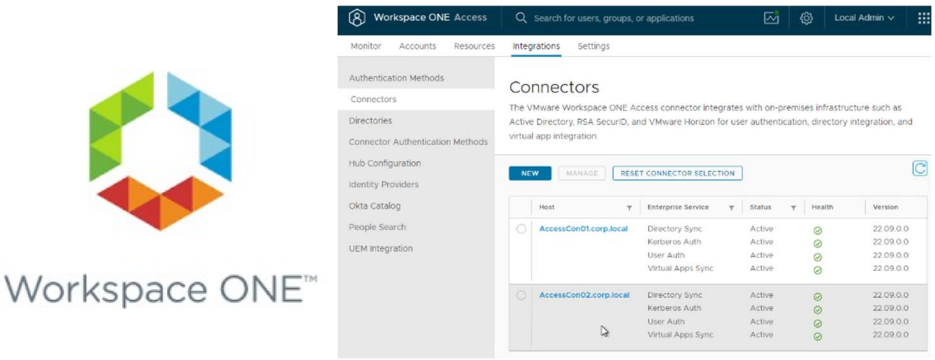

Рисунок 2.10 – VMware Workspace ONE (логотип та інтерфейс)

Інтелектуальний центр Workspace ONE забезпечує самообслуговування користувачів, підвищуючи продуктивність і мінімізуючи навантаження на IT-команди. Автоматичне управління виправленнями підтримує актуальність операційних систем і програмного забезпечення, а засоби контролю безпеки, такі як умовний доступ і шифрування даних, додатково захищають віддалені та мобільні пристрої, забезпечуючи відповідність нормативним вимогам навіть за межами корпоративної мережі.

Cisco Meraki Systems Manager. Хмарне рішення для моніторингу пристроїв та управління кінцевими точками, яке підходить для організацій, що використовують обладнання Cisco Meraki, підтримує пристрої на базі Windows, macOS, iOS та Android, а також надає центральну інформаційну панель для управління додатками та конфігурацією. Рішення дозволяє виконувати віддалені дії, такі як встановлення додатків і перезавантаження пристроїв, а також забезпечує безпеку за допомогою моніторингу в режимі реального часу і умовного доступу (див. рис. 2.11) [23].

Рисунок 2.11 – Cisco Meraki Systems Manager (логотип та інтерфейс)

Завдяки автоматичному оновленню операційної системи і додатків, Meraki Systems Manager підвищує безпеку віддалених і мобільних пристроїв, дозволяючи швидко усувати вразливості. Цей інструмент особливо підходить для користувачів мереж Cisco, яким потрібне просте і масштабоване управління пристроями.

SolarWinds N-central. Потужне рішення для управління кінцевими точками, призначене для підтримки віддаленого моніторингу, управління виправленнями та розгортання програмного забезпечення на пристроях Windows і macOS для постачальників керованих послуг (MSP) і підприємств. Адміністратори можуть віддалено усувати проблеми, встановлювати ПЗ та налаштовувати параметри. Можливості моніторингу N-central надають інформацію про стан та продуктивність пристрою в режимі реального часу, що дозволяє IT-командам виявляти проблеми та їх оперативно знешкоджувати. (див. рис. 2.12) [23].

Рисунок 2.12 – SolarWinds N-central (логотип та інтерфейс)

Автоматичне управління виправленнями забезпечує безперервне оновлення операційної системи і додатків, усуваючи вразливості на пристроях за межами корпоративної мережі. n-central інтегрований з іншими інструментами SolarWinds, що дозволяє легко задовольнити потреби в управлінні мережею і пристроями. n-central інтегрований з іншими інструментами SolarWinds, що полегшує задоволення потреб в управлінні мережею та пристроями.

ManageEngine Endpoint Central. Рішення для управління кінцевими точками, що охоплює ноутбуки, настільні комп’ютери, мобільні пристрої та сервери, яке автоматизує розгортання програмного забезпечення, оновлення операційної

системи та додатків, а також впровадження політики безпеки. IT-команди виконують віддалені дії, такі як перезавантаження, встановлення та видалення програм, що дозволяє швидко знаходити та вирішувати проблеми. Як і в SolarWinds, стан пристроїв відстежується в режимі реального часу, що дозволяє адміністраторам виявляти проблеми з продуктивністю та загрози безпеці. Якщо на пристрої виявлено підозрілу активність, Endpoint Central може обмежити доступ до ресурсів компанії до вирішення проблеми (див. рис. 2.13) [23].

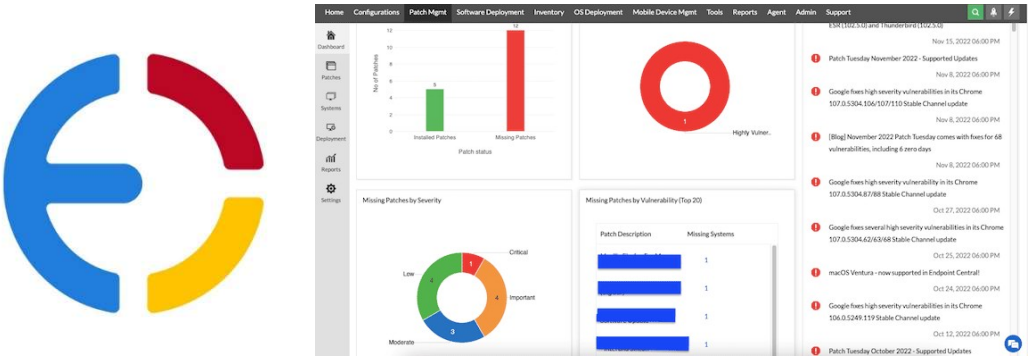

Рисунок 2.13 – ManageEngine Endpoint Central (логотип та інтерфейс)

Функція управління виправленнями є високоавтоматизованою, оновлює і підтримує операційні системи і додатки, а також своєчасно усуває вразливості в системі безпеки. Це особливо важливо для віддалених пристроїв за межами корпоративної мережі, які потребують постійного захисту, незважаючи на ненадійні мережеві з'єднання.

Jamf Pro. Провідне рішення для управління пристроями Apple, яке надає кастомну функціональність для macOS, iOS та tvOS. Система спрощує управління кінцевими точками Apple, включаючи розгортання додатків, застосування політики безпеки та оновлення системи, дозволяючи IT-командам віддалено керувати конфігурацією, встановлювати та оновлювати додатки, а також контролювати системні налаштування. Завдяки таким функціям, як розгортання «без дотику», Jamf Pro ідеально підходить для компаній, які значною мірою покладаються на пристрої Apple, забезпечуючи узгоджену конфігурацію та безпеку в організації (див. рис. 2.14) [23].

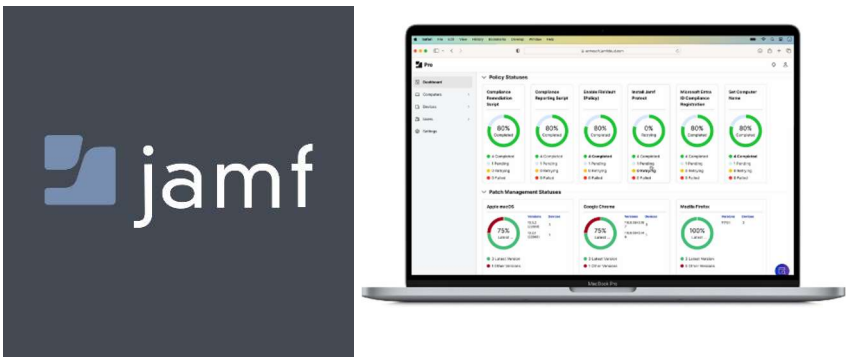

Рисунок 2.14 – Jamf Pro (логотип та інтерфейс)

Автоматизоване управління виправленнями для macOS та iOS у Jamf Pro усуває вразливості та забезпечує безпеку систем завдяки своєчасному оновленню ОС та програмного забезпечення. Інструмент ідеально підходить для управління пристроями Apple як у корпоративному, так і у віддаленому робочому середовищі, оскільки він може віддалено виконувати такі дії з безпеки, як блокування та очищення пристроїв у разі крадіжки.

Ivanti Unified Endpoint Manager. Потужне рішення для управління кінцевими точками, яке забезпечує управління виправленнями, віддалене управління і розгортання програмного забезпечення для настільних комп'ютерів, ноутбуків і мобільних пристроїв. З акцентом на безпеку і відповідність вимогам, Ivanti ідеально підходить для управління пристроями Windows, macOS, iOS і Android в середовищах, де потрібна сувора відповідність вимогам. IT-команди можуть вживати заходів віддалено і контролювати продуктивність пристроїв і стан системи (див. рис. 2.15) [23].

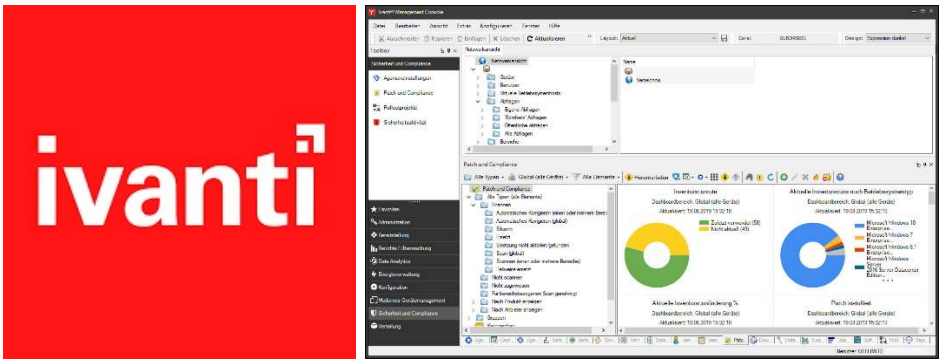

Рисунок 2.15 – Ivanti Unified Endpoint Manager (логотип та інтерфейс)

Функція автоматичного оновлення Ivanti підвищує безпеку пристроїв за межами корпоративної мережі, гарантуючи, що кінцеві точки завжди оновлені. Адміністратори можуть швидко реагувати на загрози та проблеми з продуктивністю і, за необхідності, заблокувати корпоративні ресурси, щоб запобігти несанкціонованому доступу.

Citrix Endpoint Management. Уніфіковане рішення для керування як мобільними, так і традиційними десктопами, що легко інтегрується з віртуальними додатками та десктопами Citrix. Підтримка iOS, Android, Windows і macOS, розгортання додатків, моніторинг пристроїв, політики відповідності. Адміністратори можуть віддалено виконувати такі дії, як перезапуск або оновлення програм з будь-якого місця, а також впроваджувати безпекові рішення по захисту на пристроях (див. рис. 2.16) [23].

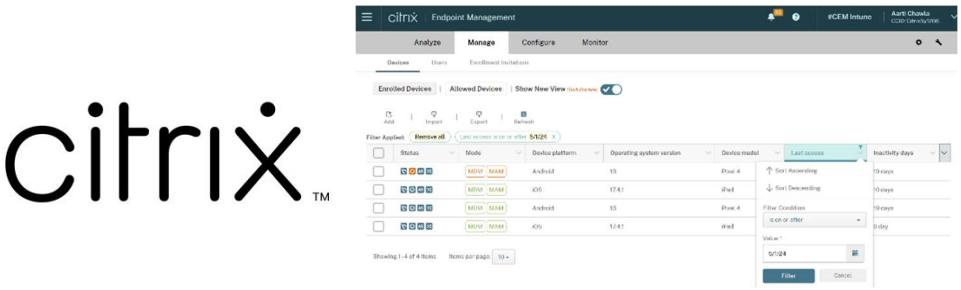

Рисунок 2.16 – Citrix Endpoint Management (логотип та інтерфейс)

Функція автоматичного встановлення виправлень у Citrix Endpoint Management гарантує, що операційні системи та додатки будуть оновлюватися, забезпечуючи безперервний захист як корпоративних, так і віддалених пристроїв. Тісна інтеграція з віртуалізованими середовищами робить Citrix ідеальним рішенням для управління віддаленими працівниками.

Sophos Mobile. Орієнтоване на безпеку рішення для управління мобільними пристроями (MDM), яке забезпечує безпеку кінцевих точок, управління додатками та відповідність вимогам для пристроїв iOS, Android та Windows 10. Sophos Mobile дозволяє виконувати такі дії з безпеки, як віддалена конфігурація, моніторинг пристроїв, а також блокування та очищення пристроїв у разі втрати або крадіжки.

Адміністратори також можуть контролювати відповідність вимогам і керувати додатками для захисту кінцевих точок (див. рис. 2.17) [23].

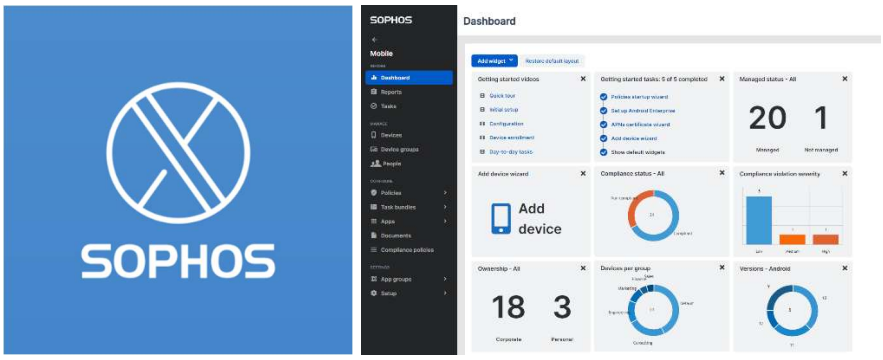

Рисунок 2.17 – Sophos Mobile (логотип та інтерфейс)

Автоматичне управління виправленнями Sophos Mobile надає регулярні оновлення для усунення вразливостей і захисту пристроїв, де б вони не знаходилися. Як рішення, орієнтоване на безпеку, воно ідеально підходить для організацій з мобільною робочою силою, де захист даних є пріоритетом.

BlackBerry UEM. Інтегроване рішення для управління кінцевими точками, орієнтоване на безпеку, яке підходить для управління мобільними та настільними пристроями, включаючи iOS, Android, Windows і macOS. Забезпечуючи управління додатками, відстеження пристроїв і дотримання вимог безпеки для регульованих середовищ, BlackBerry UEM особливо корисний в галузях із суворими вимогами до захисту даних, оскільки він може віддалено виконувати дії з безпеки, такі як блокування і очищення пристроїв у разі втрати (див. рис. 2.18) [23].

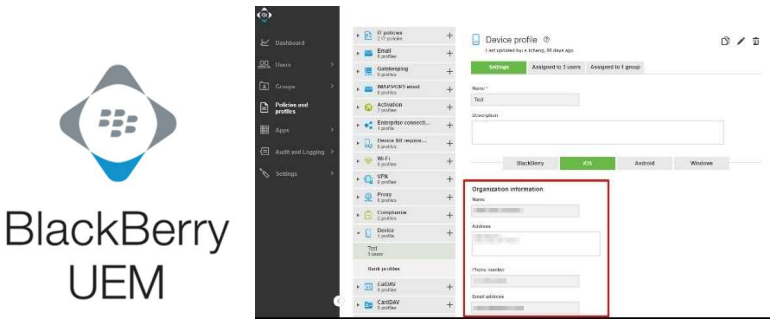

Рисунок 2.18 – BlackBerry UEM (логотип та інтерфейс)

Автоматизовані процеси встановлення виправлень забезпечують актуальність операційних систем і додатків та безпеку віддалених пристроїв, а потужні функції комплаєнсу та безпеки BlackBerry UEM ідеально підходять для фінансового, медичного та державного секторів.

Hexnode UEM. Одне з найбільш універсальних рішень для управління кінцевими точками з підтримкою iOS, Android, Windows, macOS і Chrome OS. Надає інструменти для розгортання додатків, віддаленого керування та дотримання вимог безпеки. IT-команди можуть застосовувати політики безпеки, встановлювати ПЗ та відстежувати стан пристроїв у режимі реального часу. (див. рис. 2.19) [23].

Рисунок 2.19 – Hexnode UEM (логотип та інтерфейс)

Автоматизоване управління виправленнями гарантує, що операційні системи і додатки будуть оновлюватися, а віддалені пристрої залишатимуться в безпеці. Hexnode простий і економічно ефективний, що робить його популярним вибором для підприємств з обмеженим бюджетом. Baramundi Management Pack

Baramundi Management Suite. Німецьке рішення для управління кінцевими точками, яке підтримує всі відомі ОС. Відоме своїми потужними функціями відповідності вимогам, воно пропонує управління виправленнями, відстеження інвентаризації та віддалене управління і підходить для регульованих середовищ. Адміністратори можуть віддалено встановлювати ПЗ, налаштовувати пристрої та застосовувати політики безпеки (див. рис. 2.20) [23].

Рисунок 2.20 – Baramundi Management Suite
(логотип та інтерфейс)

Можливості автоматизованого встановлення виправлень гарантують швидке усунення вразливостей і захист кінцевих точок від загроз. Функції звітності та моніторингу та підходять для компаній із суворими регуляторними вимогами.

SOTI MobiControl. Підтримує iOS, Android, Windows і Linux і призначений для управління надійними пристроями Інтернету речей в таких галузях, як роздрібна торгівля і логістика. Він пропонує віддалене управління, розгортання і моніторинг додатків, а також віддалене усунення несправностей і налаштування (див. рис. 2.21) [23].

Рисунок 2.21 – SOTI MobiControl (логотип та інтерфейс)

Автоматичні оновлення ОС і додатків забезпечують безпеку пристроїв за межами корпоративної мережі, а SOTI MobiControl особливо корисний для управління спеціалізованими пристроями в суворих умовах, забезпечуючи безпеку і управління, які відповідають специфічним потребам галузі.

Kaseya VSA. Розроблений для постачальників керованих послуг (MSP) та IT-команди, Kaseya VSA підтримує віддалений моніторинг, управління виправленнями та розгортання програмного забезпечення для Windows, macOS та мобільних пристроїв. Kaseya VSA пропонує можливість віддаленого усунення несправностей, налаштування та застосування політик безпеки на пристроях за допомогою централізованої консолі (див. рис. 2.22) [23].

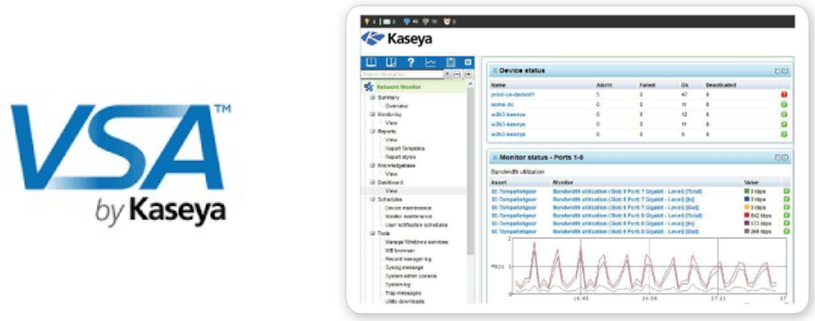

Рисунок 2.22 – Kaseya VSA (логотип та інтерфейс)

Автоматизоване управління виправленнями допомагає захистити кінцеві точки, підтримуючи операційні системи і додатки в актуальному стані. Це особливо важливо для віддалених і мобільних пристроїв, які потребують захисту за межами корпоративної мережі, оскільки Kaseya VSA інтегрується з іншими рішеннями Kaseya і адаптується до складних IT-середовищ.

Модуль довіреної платформи (TPM). Апаратним рішенням, яке відіграє ключову роль у забезпеченні безпеки системи на апаратному рівні, є Trusted Platform Module (TPM) - апаратний модуль безпеки, вбудований у більшість сучасних ПК та ноутбуків. TPM використовуються для шифрування даних, моніторингу стану системи та захисту від несанкціонованих вторгнень на апаратному рівні (див. рис. 2.23) [23].

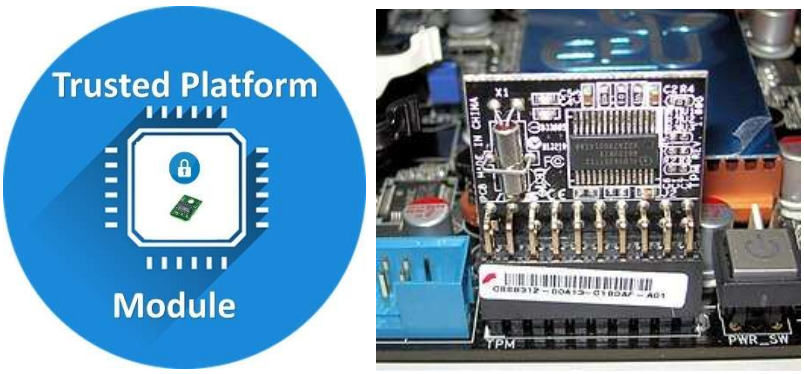

Рисунок 2.23 – Модуль довіреної платформи (TPM) (логотип та інтерфейс)

Одним із прикладів використання TPM є підтримка технології шифрування диска, я BitLocker. TPM зберігає ключі шифрування, що дозволяє захистити дані навіть у разі втрати або крадіжки пристрою – це є надзвичайно важливим для віддалених працівників, чиї пристрої часто знаходяться поза фізичним контролем організацій, але варто зазначити, що втрата ключів шифрування призведе до втрати даних.

Крім шифрування, TPM забезпечує захист цілісності системи на етапі завантаження – модуль перевіряє стан пристрою на наявність несанкціонованих змін і попереджає про можливі спроби втручання. TPM дозволяє адміністраторам моніторити стан кінцевих пристроїв та здійснювати контроль над процесами шифрування й аутентифікації на апаратному рівні, що значно знижує ризики компрометації даних. Але найголовніше стосовно TPM – пропонує лише потужні функції безпеки, він не є комплексним інструментом моніторингу. Він головним чином зосереджений на безпеці пристрою, а не на продуктивності чи діагностичному моніторингу, тому він доповнює, а не замінює інші інструменти апаратного забезпечення повністю.

2.4 Застосування IoT для дистанційного контролю кінцевих пристроїв

Окрім апаратних та програмних засобів для моніторингу, важливу роль сьогодні відіграють рішення, засновані на технологіях Інтернету речей (IoT, Internet of Things). Ці технології активно використовуються у наших містах, будинках і на робочих місцях, зокрема для дистанційного моніторингу кінцевих пристроїв, що є особливо важливим для працівників, які працюють віддалено.

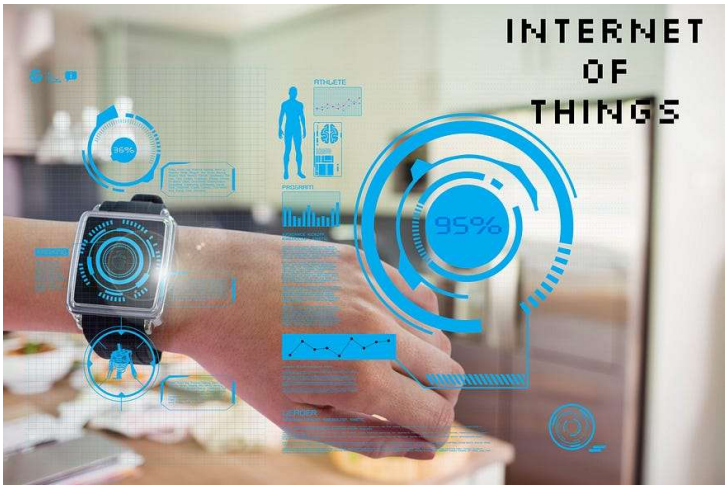

Рисунок 2.24 – Інтеграція Інтернету речей для моніторинку контролю пристроїв

Інтеграція IoT-сенсорів та платформ для обробки даних дозволяє забезпечити постійний контроль за станом пристроїв. Завдяки цьому можна збирати дані про їхню продуктивність, виявляти можливі несправності або відхилення у роботі та оперативно на них реагувати, що дозволить не лише моніторити роботу віддалених пристроїв, а також підвищить ефективність їхнього використання, зменшуючи ризики простоїв та забезпечуючи стабільну роботу навіть у віддалених умовах. Технології IoT стають невід'ємною частиною сучасних рішень для віддаленого управління й моніторингу, роблячи процеси більш ефективними та надійними (див. рис. 2.24) [24].

Більш детальний погляд на використання датчиків для збору даних в контексті Інтернету речей (IoT) виявляє кілька важливих аспектів:

— контроль температури;

— моніторинг енергоспоживання;

— аналіз роботи системи.

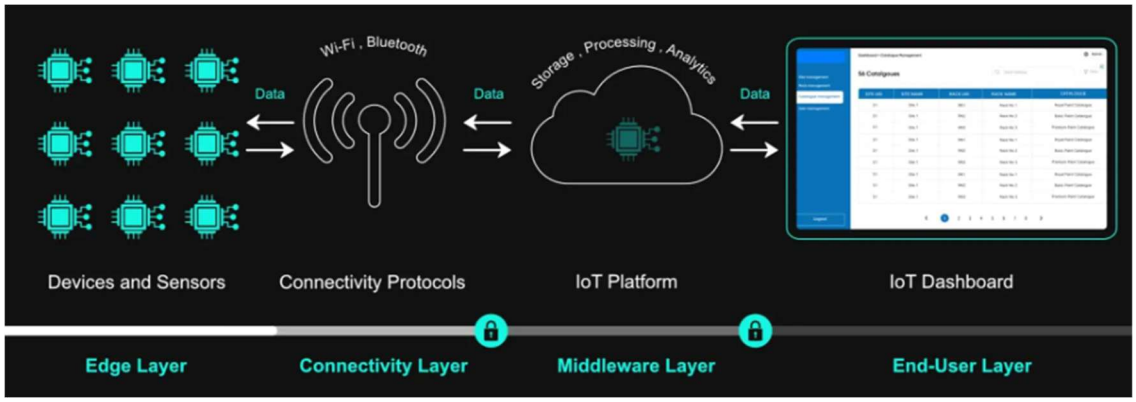

Рисунок 2.25 – Отримання даних з IoT сенсорів до «дашборду»

Контроль температури. Один з основних прикладів – це контроль температурного режиму. Сенсори дозволяють моніторити температуру процесора чи інших компонентів системи. Якщо температура перевищує допустимий рівень, система може автоматично сповістити технічну підтримку або працівника про необхідність вжиття заходів, наприклад, зниження навантаження чи перевірки пристрою. Такий засіб особливо важливий в умовах інтенсивної роботи пристроїв, таких як ноутбуки або ПК, де сенсори можуть виявити перегрівання і автоматично знижувати продуктивність, щоб уникнути пошкоджень компонентів (див. рис. 2.25) [24, 25].

Проте, подібні завдання можуть виконуватись і без використання IoT. Наприклад, в ручному режимі або за допомогою програмного забезпечення можна зменшити частоту процесора, тим самим знизивши його нагрівання, що особливо актуально для застарілого обладнання, де перегрів можна компенсувати шляхом зменшення продуктивності. Такі технології, як Intel Turbo Boost, дозволяють тимчасово збільшити частоту процесора для підвищення продуктивності, але це також приведе до тротлінгу. У таких випадках Turbo Boost можна вимкнути, або ж

програмно зменшити навантаження на процесор, що допоможе підтримувати прийнятну температуру.

Моніторинг енергоспоживання. Ще один важливий момент – моніторинг енергоспоживання. Датчики можуть фіксувати енергоспоживання обладнання та допомагають виявити необґрунтоване споживання або перевантаження електромережі, що особливо корисно для мобільних пристроїв і ноутбуків, які працюють від батарей. Система може попередити користувача про наближення розрядки або запропонувати оптимізувати налаштування для економії енергії.

Альтернативою використанню IoT-рішень можуть бути вбудовані можливості операційних систем, наприклад, Windows, де є режим енергоефективності, що дозволяє знижувати навантаження на систему за рахунок відключення фонових процесів. Крім того, виробники, такі як ASUS, пропонують спеціальні рішення для управління енергоспоживанням, як режим «Еко» в програмі Armoury Crate, який дозволяє відключити дискретну відеокарту на ігрових ноутбуках для продовження часу роботи від батареї (див. рис. 2.26).

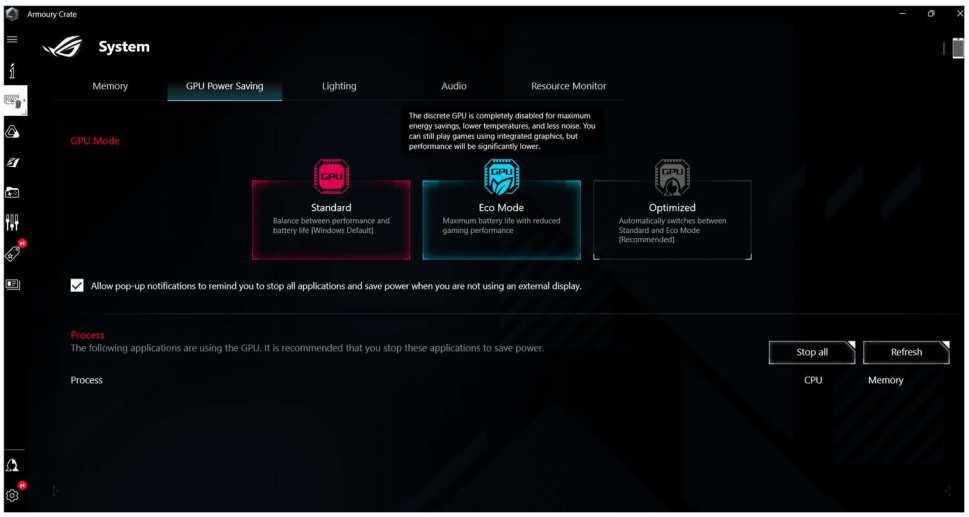

Рисунок 2.26 – Режим «Еко» для максимальної автономності (Armoury Crate)

Аналіз роботи системи. Сенсори також здатні збирати дані про роботу системи, контролюючи навантаження на процесор, оперативну пам’ять, швидкість мережі та використання дисків [25]. Даний засіб дозволить IT-відділам швидко вирішувати проблеми продуктивності та надавати рекомендації користувачам

щодо оптимізації роботи. Наприклад, при виявленні аномальної активності процесора або підозрілого використання системних ресурсів, сенсори можуть сповістити користувача та заблокувати небажані процеси, що допоможе підтримувати стабільну та ефективну роботу системи.

Всі розглянуті рішення для збору даних про стан пристроїв за допомогою IoT-сенсорів виконують завдання моніторингу, збору інформації та передачі її адміністраторам для подальшого управління. Йдеться про такі аспекти, як контроль температурних режимів, енергоспоживання та аналіз роботи систем. Важливо мати платформи для їхньої подальшої обробки, які дозволяють централізовано моніторити та ефективно управляти кінцевими пристроями, зокрема віддаленими (див. рис. 1.5, 2.25, 2.26) [5, 24, 25].

Платформи для обробки даних — це важливий елемент IoT-систем, що забезпечують моніторинг та автоматизацію процесів на основі зібраних даних. Вони дозволяють не лише збирати інформацію, але й обробляти її та запускати різноманітні дії на основі аналітики. Приклади таких платформ — AWS IoT Core та Azure IoT Hub, які надають інструменти для інтеграції пристроїв і автоматичного реагування на події (див. рис. 2.27) [26, 27].

Рисунок 2.27 – Платформи для обробки даних з IoT сенсорів (AWS IoT Core та Azure IoT Hub)

AWS IoT Core. Надає різним пристроям підключатися до хмарної платформи для збору даних. Наприклад, кінцеві пристрої, такі як ноутбуки, оснащені датчиками, можуть надсилати на платформу AWS дані про стан системи, такі як

температура та використання ресурсів. Ці дані можуть бути автоматично проаналізовані, і в разі виявлення перегріву або перевантаження системи можуть бути вжиті заходи для оптимізації продуктивності пристрою, наприклад, автоматичне зниження продуктивності процесора або надсилання повідомлень про необхідність проведення технічних перевірок. [26, 27].

AWS IoT Core також дозволяє обробляти великі обсяги даних з тисяч кінцевих пристроїв, що робить його придатним для масштабованих рішень великих організацій. Віддалені працівники можуть отримувати автоматичні оновлення та рекомендації для покращення продуктивності своїх пристроїв на основі аналізу отриманих даних. Наприклад, якщо сенсори виявляють аномальні дії з боку пристрою, такі як надмірне використання ресурсів або підозріла активність у мережі, платформа може автоматично запустити сценарій для тимчасового відключення пристрою від корпоративних ресурсів [26, 27]..

Azure IoT Hub. Забезпечує аналогічну функціональність, дозволяючи централізовано керувати кінцевими пристроями з двостороннім зв’язком між ними та хмарною платформою. Таке рішення дозволяє адміністраторам отримувати реальні дані про стан систем, автоматизувати процеси технічного обслуговування та оперативно реагувати на проблеми. Однією з ключових можливостей Azure IoT Hub є аналітика та прогнозування. Платформа дозволяє аналізувати дані та застерігати про можливі збої в роботі пристроїв. Наприклад, на основі аналізу температурного режиму система може прогнозувати ризики перегріву та рекомендувати користувачу вжити профілактичних заходів, таких як заміна компонентів системи охолодження, дозволить уникнути серйозних несправностей і підтримувати стабільну роботу пристроїв [27].

Автоматизація та сценарії реагування. Як AWS IoT Core, так і Azure IoT Hub дозволяють автоматизувати запуск різних процесів на основі зібраних даних. Наприклад, якщо виявлено перевищення допустимого навантаження на процесор, система може автоматично знизити продуктивність без втручання користувача. Якщо ж сенсори виявляють надмірну кількість запитів до корпоративної мережі з

боку кінцевого пристрою, платформа може запустити сценарій для тимчасового відключення цього пристрою від мережі, доки ситуація не буде вирішена [26, 27]..

Масштабованість рішень. І AWS, і Microsoft надають рішення, здатні обробляти великі обсяги даних від великої кількості пристроїв, що є ключовим для великих організацій. Масштабованість таких платформ дозволяє віддаленим працівникам отримувати постійний моніторинг та автоматичні рекомендації щодо покращення продуктивності їх пристроїв, що забезпечує стабільну та ефективну роботу навіть за великих навантажень [26, 27].

Якщо говорити про практичні приклади використання IoT (інтернету речей) для обслуговування пристроїв, які використовуються віддаленими працівниками, можна розглянути кілька прикладів.

Сповіщення про перевантаження компонентів пристрою. Якщо один з компонентів кінцевого пристрою, наприклад, ноутбук або стаціонарний комп'ютер, виявляє перевантаження, система може автоматично надіслати сповіщення на основі даних з датчиків Інтернету речей. Наприклад, повідомлення може інформувати користувача про потребу вирішення будь-якого моменту задля оптимізації продуктивності пристрою, наприклад, перезавантажити або зменшити навантаження на систему. Система може автоматично знизити продуктивність пристрою, запустивши відповідні сценарії оптимізації, або надіслати користувачеві пропозицію завершити неактивні програми та звільнити системні ресурси, таким чином уникаючи серйозних проблем і підтримуючи стабільну роботу пристрою навіть під час інтенсивного використання [26, 27].

Управління енергоспоживанням. Для мобільних пристроїв, таких як ноутбуки, IoT-сенсори можуть моніторити рівень використання батареї. На основі зібраних даних платформа може надсилати користувачеві рекомендації щодо економії енергії, наприклад, перемикати пристрій у режим економії або оптимізувати налаштування для продовження роботи без підзарядки. Також система може автоматично вимикати невикористовувані компоненти, такі як дискретна відеокарта, або знижувати яскравість екрану для збереження заряду батареї. Наприклад, рішення, подібне до Armoury Crate від ASUS, може

автоматично перемикати пристрій у deko-режим при відключенні від зарядки, що збільшує час автономної роботи ноутбука до 6-8 годин, замість звичайних 2-3 годин (див. рис. 2.26).

Моніторинг мережевого підключення. IoT-сенсори також можуть відстежувати якість мережевого з'єднання віддалених працівників. У разі виявлення нестабільного інтернет-з'єднання, система може автоматично оптимізувати мережеві налаштування або пропонувати альтернативні рішення для покращення зв'язку. Наприклад, якщо працівник стикається з проблемами підключення до основної мережі, система може запропонувати використання мобільної мережі або іншої доступної мережі для забезпечення стабільної роботи [1, 4, 5, 24].

2.5 Автоматизація технічного обслуговування за допомогою скриптів і віддаленого доступу

Автоматизація технічного обслуговування неможлива без двох ключових елементів: сценаріїв (скриптів) та віддаленого доступу. Особливо це важливо для роботи з віддаленими працівниками, до пристроїв яких необхідно мати доступ для виконання завдань без їхнього прямого втручання. Без цих двох компонентів автоматизація технічного обслуговування стає малоефективною.

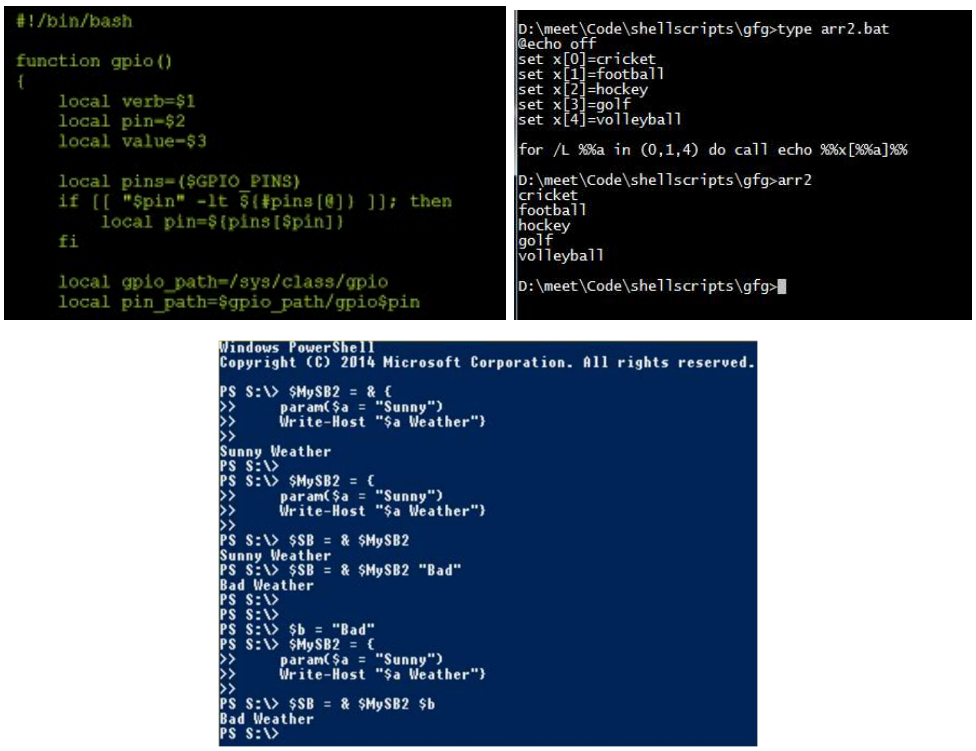

Рисунок 2.28 – Сценарії (скрипти) автоматизації рутинних процесів (Bash, Cmd, PowerShell)

Скрипти є основою для автоматизації, оскільки вони дозволяють IT-фахівцям створювати сценарії для виконання рутинних завдань, таких як діагностика, оновлення або налаштування пристроїв, які забезпечать можливість швидкого й ефективного обслуговування кінцевих пристроїв, які використовуються віддаленими працівниками, без потреби фізичної присутності IT-спеціаліста або без втручання самого працівника (див. рис. 2.28) [28].

Віддалений доступ є другим ключовим елементом, оскільки він дозволяє виконувати всі необхідні дії, незалежно від того, де знаходиться пристрій. Завдяки цьому IT-фахівці можуть оперативно реагувати на технічні проблеми, не витрачаючи час на фізичний доступ до пристрою, що також важливо для забезпечення безперервної роботи віддалених працівників.

Для автоматизації технічного обслуговування найчастіше використовують кілька мов для написання скриптів:

— PowerShell (Windows, частково Linux);

— Bash (Linux, macOS);

— Python (повноцінна мова програмування).

PowerShell. Починаючи з PowerShell, оскільки це один із найбільш потужних інструментів для адміністрування Windows-систем. PowerShell є оновленою версією командної оболонки для Windows, яка розроблена для автоматизації різноманітних задач [28]. Він дозволяє IT-фахівцям не лише керувати локальними пристроями, але й використовувати його на системах Linux, що робить PowerShell універсальним засобом для автоматизації. PowerShell дозволяє виконувати складні завдання через написання сценаріїв (скриптів). Наприклад, за допомогою скриптів можна автоматизувати такі завдання, як видалення тимчасових файлів для звільнення місця на диску.

Проте тут важливо правильно налаштувати скрипти, щоб вони не видаляли системні або приховані файли, які є критично важливими для роботи операційної системи. У Windows є певні приховані файли, наприклад, на робочому столі існує невидимий файл «desktop.ini», який забезпечує правильну роботу цієї папки. Видалення таких файлів може порушити роботу системи. Тому в скриптах PowerShell можна задати винятки для збереження системних файлів і видалення лише тимчасових або небажаних даних.

За допомогою PowerShell можна автоматизувати такі процеси, як перезавантаження пристроїв, налаштування автоматичного включення та виключення комп’ютерів або інші рутинні завдання. Наприклад, для дистанційного включення комп’ютера можна налаштувати функцію Wake on LAN, яка дозволяє вмикати пристрої через локальну мережу. Однак, для використання цієї функції віддалено необхідно налаштувати перенаправлення портів на маршрутизаторі та налаштувати VPN.

Також PowerShell дозволяє виконувати регулярні перевірки стану системи. Наприклад, можна створити скрипт для перевірки стану дисків або наявності пошкоджених секторів. Для цього існують спеціальні командлети, такі як «Get-Volume» або «Test-Volume», які фактично є вдосконаленими версіями старих команд, таких як «chkdsk». PowerShell також дозволяє налаштувати систему повідомлень про виявлені проблеми. Як приклад, це можуть бути повідомлення

через електронну пошту або системи моніторингу, які автоматично надсилають оповіщення IT-фахівцям про помилки або зниження продуктивності [28].

Bash. Аналогічно PowerShell, Bash є потужним інструментом для автоматизації в Linux-системах. За допомогою скриптів на Bash можна автоматизувати оновлення системи через пакетні менеджери (apt, yum, dnf), перевірку системних журналів за допомогою команд grep та моніторинг системних ресурсів (команди top, df, vmstat). Ці можливості дозволяють IT-фахівцям стежити за станом системи та виконувати рутинні завдання без втручання користувача [28].

Python. Підходить для автоматизації; завдяки таким бібліотекам, як psutil, Python може збирати та обробляти системні дані, корисні для моніторингу віддалених пристроїв. Крім того, API-інтерфейси Python можна використовувати для автоматизації розгортання програмного забезпечення на таких платформах, як Microsoft Intune і Google Workspace, а також для налаштування обміну повідомленнями через такі сервіси, як Slack і Telegram [14, 18, 19, 29].

У Windows існують потужні пакетні менеджери, такі як «Winget» і «Chocolatey», які значно спрощують процес автоматизації встановлення та оновлення програмного забезпечення на пристроях, що особливо корисно для віддалених працівників, оскільки дозволяє ефективно керувати програмами без необхідності написання складних скриптів з нуля. Водночас, скрипти автоматизації можуть інтегрувати використання цих пакетних менеджерів, надаючи більш компактне рішення для конкретних завдань.

Пакетний менеджер «Winget». Winget (Windows Package Manager) є універсальним інструментом для автоматизації процесів, пов’язаних зі встановленням, оновленням та видаленням програм на Windows-пристроях. Він забезпечує значну зручність, дозволяючи встановлювати необхідні програми, такі як браузери, VPN-клієнти або офісні пакети, через єдину команду «winget install». Наприклад, якщо необхідно оновити все встановлене ПЗ, можна виконати команду «winget upgrade –all», що автоматично оновить усі наявні програми, навіть якщо вони були встановлені раніше без використання winget. Winget також може використовуватись для видалення попередніх версій програм перед їх оновленням,

що особливо корисно при необхідності переходу на нові версії програмного забезпечення (див. рис. 2.29) [30].

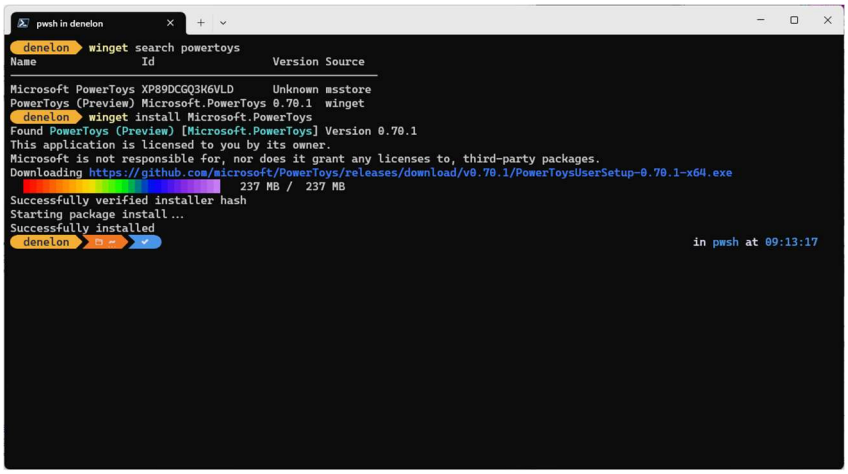

Рисунок 2.29 – Пакетний менеджер Winget (Windows Package Manager)

Пакетний менеджер «Chocolatey». Chocolatey є більш універсальним та розширеним пакетним менеджером для Windows, який надає широкий набір функцій для керування програмами. Він не лише спрощує процеси встановлення, видалення та оновлення програм через командний рядок або скрипти, але й інтегрується з процесами Continuous Integration/Continuous Deployment (CI/CD), роблячи Chocolatey корисним інструментом для автоматизації розгортання програмного забезпечення на кінцевих пристроях. Однією з ключових функцій Chocolatey є можливість створення резервних копій встановлених пакетів, що дозволить швидко відновлювати конфігурації пристроїв у разі збою або перевстановлення системи (див. рис. 2.30) [31].

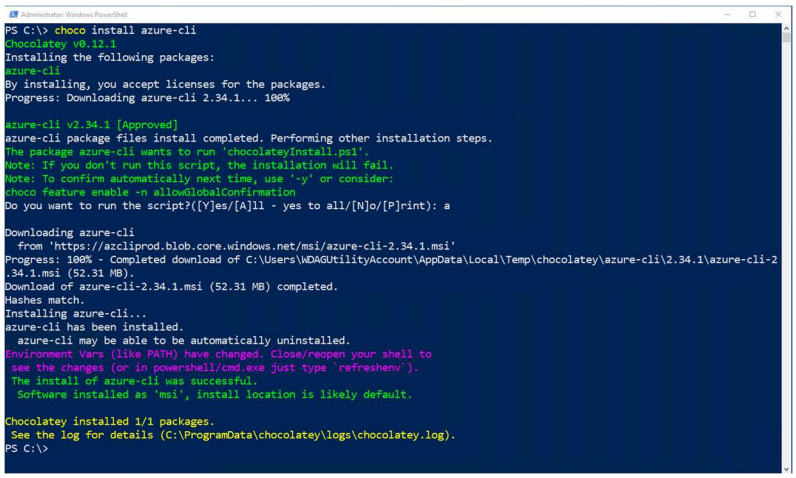

Рисунок 2.30 – Пакетний менеджер *Chocolatey*

Пакетний менеджер створює спеціальний файл журналу (лог-файл), який містить інформацію про всі встановлені програми. Цей файл можна експортувати та використовувати для відновлення програм на іншому пристрої або після перевстановлення операційної системи. Крім того, Chocolatey дозволяє легко переносити конфігурації між пристроями. Наприклад, після експорту лог-файлу з одного комп’ютера, його можна імпортувати на інший пристрій або на ту ж машину після перевстановлення системи, що значно полегшує процес відновлення програмного середовища, оскільки всі програми, що були встановлені через Chocolatey, можуть бути швидко повернуті.

Як «Winget», так і «Chocolatey» дозволяють керувати широким спектром програм, однак їхні можливості обмежені набором програм, які доступні в їх базах даних [43]. Winget здебільшого орієнтується на завантаження програм з Microsoft Store або інших сховищ, пов’язаних з GitHub. Якщо певні програми, наприклад специфічні драйвери або програми, не знаходяться у цих базах, їх не можна буде автоматично відновити чи встановити через ці пакетні менеджери [29, 30].

Автоматизація за допомогою скриптів є надзвичайно важливим аспектом у багатьох сферах діяльності, оскільки вона дозволяє автоматизувати складні процеси, спрощуючи рутинні завдання та забезпечуючи стабільність роботи систем, і це може стосуватися не тільки Windows, а й інших операційних систем і

технологій, де скрипти допомагають підвищувати продуктивність і оптимізувати управління різними процесами.

Особливо яскраво це проявляється в сфері DevOps, де автоматизація є ключовою складовою щоденної роботи. DevOps інтегрує розробку і операційну діяльність, і скрипти тут грають центральну роль у забезпеченні безперебійної роботи всієї системи. Наприклад, кожен день DevOps-інженери можуть створювати та інтегрувати нові патчі, налаштовувати скрипти для моніторингу, оновлення та підтримки систем. Головне, щоб усі компоненти інфраструктури працювали злагоджено, і саме скрипти дозволяють автоматизувати ці завдання.

В процесі автоматизації за допомогою скриптів можна налаштувати регулярні перевірки всіх систем, виявлення помилок та автоматичне відновлення. Наприклад, якщо виникає збій, скрипт може виявити проблему і автоматично запустити процес її усунення, дозволяючи підтримувати високу продуктивність систем і уникати тривалих простоїв. Якщо проблема все ж виникла, можна швидко відновити працездатність системи за допомогою ретельно підготовленого скрипту або навіть комплексу сценаріїв, які забезпечують гнучкість і надійність у різних ситуаціях.

Автоматизація процесів значно полегшує управління системами, але є випадки, коли необхідно використовувати додаткові інструменти для віддаленого доступу, що особливо важливо для вирішення складніших завдань, які не можуть бути виконані виключно за допомогою скриптів або автоматизованих процесів. Наприклад, коли виникають технічні проблеми на пристроях, які використовуються віддаленими працівниками, адміністратори можуть підключатися до цих пристроїв дистанційно, щоб швидко усунути несправності без необхідності фізичної присутності на місці.

Інструменти для віддаленого доступу TeamViewer та AnyDesk. Ці інструменти дозволяють швидко і безпечно підключатися до віддаленого комп’ютера для вирішення будь-яких проблем. Вони особливо корисні там, де автоматизація не може охопити всі аспекти керування пристроями. Безкоштовні версії всіх цих інструментів доступні для некомерційних цілей, але корпоративне

використання зазвичай вимагає платної підписки. (у разі використання у комерційній діяльності – блокування доступу) (див. рис. 2.31).

Рисунок 2.31 – Ручне віддалене керування (Anydesk та TeamViewer)

Windows Remote Desktop Manager. Один із потужних недооцінених інструментів для віддаленого управління, особливо у корпоративному середовищі. Він дозволяє працювати як локально, так і віддалено, надаючи доступ до робочих столів комп'ютерів у мережі організації. У деяких випадках налаштування цього інструмента може вимагати додаткових дій, щоб забезпечити віддалений доступ до пристроїв, проте його можливості є досить широкими і можуть використовуватися для вирішення складних завдань з управління системами (див. рис. 2.32) [32].

Рисунок 2.32 – Ручне віддалене керування (Windows Remote Desktop Manager)

Для використання віддаленого доступу вкрай необхідно використовувати ті чи інші методи захищеного захисту, один із яких це VPN (Virtual Private Network).

Він є важливим елементом безпечного доступу до корпоративних ресурсів із будь-якої точки світу. VPN не тільки забезпечує захищене з’єднання між працівниками та корпоративною мережею, але й дозволяє адміністраторам керувати віддаленими пристроями, що також є частиною комплексного підходу до віддаленого обслуговування. Використання VPN допомагає забезпечити безпеку передавання даних та контроль доступу до внутрішніх ресурсів компанії [8, 22].

Коли йдеться про віддалений доступ, він дозволяє вирішувати проблеми, які не можуть бути усунені за допомогою автоматизації, що стосується, наприклад, складних діагностичних робіт або тих випадків, коли потрібно вручну налаштувати певні параметри системи. Використовуючи інструменти віддаленого доступу, адміністратори можуть працювати над вирішенням таких завдань, не витрачаючи час на фізичний доступ до пристроїв.

Підсумовуючи, поєднання автоматизації через скрипти та інструменти віддаленого доступу забезпечує швидке, ефективне та безпечне технічне обслуговування кінцевих пристроїв. Автоматизовані процеси допомагають знижувати час, витрачений на обслуговування кожного пристрою, мінімізують втручання користувачів і дозволяють підтримувати системи в робочому стані. Одночасно з цим, інструменти для віддаленого доступу надають гнучкість у вирішенні складніших завдань, підвищують рівень контролю над пристроями і дозволяють швидко реагувати на будь-які проблеми, роблячи автоматизацію і віддалений доступ невід’ємними складовими ефективного обслуговування кінцевих пристроїв, які використовуються віддаленими працівниками [29, 30, 31, 32].

РОЗДІЛ 3 РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ

3.1 Архітектура системи розподіленого технічного обслуговування

Архітектура існуючих рішень розподіленої системи управління кінцевими пристроями, яка використовуються віддаленими працівниками, складається з багаторівневої інфраструктури. Ця система дає змогу організувати ефективне централізоване управління різними пристроями, такими як персональні комп'ютери, ноутбуки, смартфони та інші пристрої, що використовуються у віддаленій роботі. Архітектура побудована за моделлю «клієнт-сервер», де центральна платформа здійснює контроль і координацію обслуговування кінцевих пристроїв.

Ключові компоненти архітектури:

— централізований сервер управління;

— агенти кінцевих пристроїв (Endpoint Agents);

— база даних;

— хмарні сервіси;

— безпека та відповідність.

Централізований сервер управління. Основний командний центр, який виконує всі функції керування на сервері, де розміщені інструменти для управління кінцевими пристроями, як Mobile Device Management (MDM) та Microsoft Endpoint Configuration Manager (MECM). Сервер приймає запити на розгортання програмного забезпечення, налаштування системи, моніторинг стану пристроїв і виконання завдань з обслуговування. Централізована платформа значно спрощує процес керування, дозволяючи автоматизувати рутинні задачі, забезпечуючи

готовність інфраструктури до роботи без необхідності ручного втручання (див. рис. 3.1) [14, 17, 23, 33].

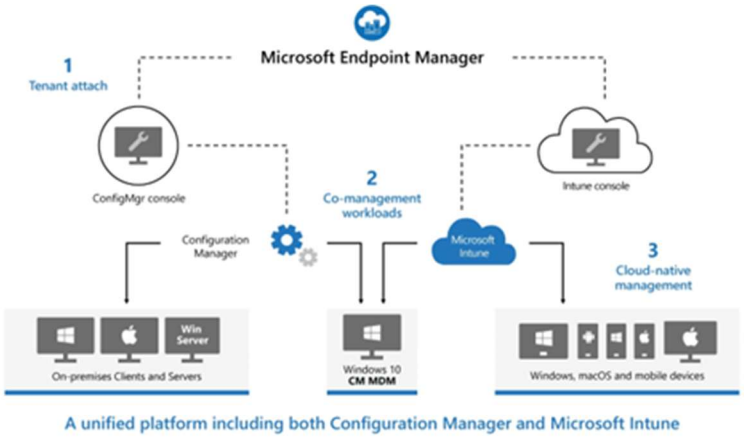

Рисунок 3.1 – Універсальна платформа на базі Microsoft Endpoint Manager (Intune + MDM+MECM)

*Агенти кінцевих пристроїв (*Endpoint Agents). На кожному пристрої (комп'ютері, смартфоні тощо) встановлюється спеціальний «агент», який працює у фоновому режимі. Він постійно взаємодіє з центральним сервером, виконуючи команди з обслуговування, наприклад, внесення змін до конфігурації чи надсилання показників продуктивності. Завдяки цьому «агенту» система може автоматично реагувати на можливі проблеми, заощаджуючи час IT-команди на вирішення технічних проблем, які користувач може навіть не помітити (див. рис. 3.2) [34].

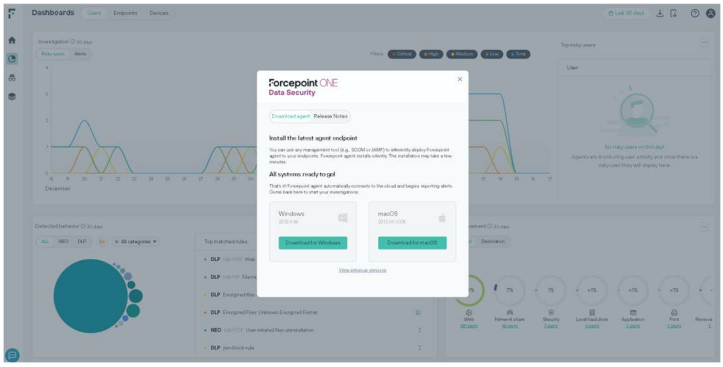

Рисунок 3.2 – Інтеграція сторонніх агентів кінцевих пристроїв *(*Endpoint Agents)

Централізована система управління допомагає мінімізувати витрати часу і ресурсів на обслуговування кінцевих пристроїв, оскільки процес моніторингу та управління автоматизований. У разі виявлення несправностей, спеціалісти IT-команди можуть швидко втрутитися, використовуючи централізований сервер як командний центр, що дозволяє уникнути ситуацій, коли віддалений працівник змушений повідомляти про проблему самостійно. Автоматизація знижує навантаження на працівників і мінімізує переривання робочого процесу.

База даних. До ключових компонентів архітектури розподіленої системи обслуговування віддалених пристроїв належить база даних, яка відіграє центральну роль у збереженні всіх необхідних даних для управління і обслуговування цих пристроїв. Без бази даних функціонування такої системи було б неможливим, оскільки саме вона забезпечує зберігання і доступ до критично важливої інформації, включаючи профілі пристроїв, конфігураційні параметри, статус відповідності політикам безпеки та журнали обслуговування, що дозволить системі постійно мати повну картину стану всіх пристроїв, забезпечуючи швидкий доступ до даних для управління і технічної підтримки.

Сучасні бази даних, такі як MongoDB та інші подібні рішення, дозволяють зберігати великий обсяг інформації та забезпечувати її безпечне зберігання і доступність в межах централізованої архітектури. Використання централізованої бази даних для автоматизованої системи обслуговування пристроїв є ключовим для уникнення можливих помилок або упущень при управлінні великою кількістю кінцевих пристроїв. У випадках, коли відсутні централізовані рішення, дані можуть зберігатися у різних місцях, що створює ризик втрати частини інформації та ускладнює аналіз (див. рис. 3.3) [35].

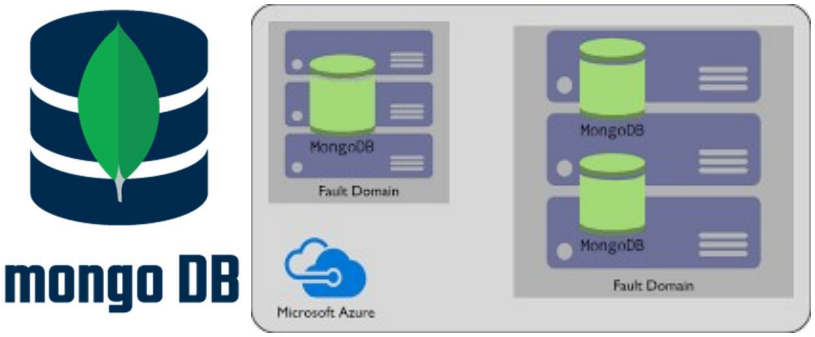

Рисунок 3.3 – Інтеграція бази даних MongoDB

Хмарні сервіси. Наступним важливим компонентом архітектури розподіленої системи обслуговування кінцевих пристроїв, які використовуються віддаленими працівниками, є інтеграція з хмарними сервісами. У сучасних умовах, де більшість бізнес-процесів пов'язана з хмарними технологіями, наявність таких сервісів є важливою умовою для забезпечення надійності, масштабованості та доступності системи. Інтеграція з хмарою дає змогу не тільки оптимізувати дистанційне управління пристроями, а й створює зручне середовище для самих працівників, яке дозволяє зберігати, оновлювати і відновлювати робочі дані без ризику їх втрати.

Хмарні платформи, такі як AWS, Google Cloud та Microsoft Azure пропонують численні можливості для зберігання, резервування та автоматичного відновлення даних – ці сервіси забезпечують регулярне резервне копіювання, яке зберігає всі необхідні налаштування та робочі матеріали користувачів, що дозволяє швидко відновити робоче середовище у разі втрати пристрою або технічної несправності. Наприклад, якщо віддалений працівник втрачає ноутбук або інший пристрій, інтеграція з хмарою дозволяє відновити його робоче середовище та дані на новому пристрої, як це працює з резервними копіями смартфонів у Apple або Google, так само, як і у випадку з переходом на новий мобільний пристрій, де резервна копія дозволяє миттєво перенести всі налаштування, додатки і дані, хмарні сервіси для корпоративних потреб забезпечують збереження всіх необхідних параметрів для автоматичного відновлення робочих середовищ, що не лише спрощує роботу IT-команди, а й дозволяє працівникам швидко повернутися до робочого процесу [14, 17, 18, 19, 21, 23, 33].

Також хмарні рішення відкривають додаткові можливості для зберігання офіційних даних у закритій корпоративній мережі, де використовується VPN і інші засоби для захисту даних від витоку. У цьому контексті інтеграція з хмарними сервісами значно підвищує рівень безпеки та стабільності системи, створюючи надійне середовище для віддаленої роботи і дозволяючи як обслуговування, так і гнучке масштабування ресурсів за потребою [8, 14, 19, 20, 33].

У базах даних також зберігаються історичні дані щодо змін, які сталися з пристроями, аналогічно до того, як у розробці програмного забезпечення використовуються інструменти для контролю версій, наприклад GitHub – це дозволяє відстежувати всі зміни в конфігураціях або статусах, надаючи змогу IT-командам бачити, коли і які дії виконувались із пристроями, а також надавати коментарі або примітки до них (див. рис. 3.4, 3.5) [30, 31, 36, 37]. (Додаток А)

Альтренатива використанню платних систем – власні сценарії автоматизації. Більшість існуючих рішень для централізованого та автоматизованого технічного обслуговування, таких як Microsoft Endpoint Configuration Manager або Mobile Device Management, що включають централізований сервер управління, агенти кінцевих пристроїв, бази даних та сервіси безпеки, є платними.

Їх впровадження вимагає від компаній значних фінансових витрат. Однак, використовуючи стандартні засоби ОС, можна створювати скрипти автоматизації, що дає значну перевагу. З одного боку, це дозволяє використовувати власні написані програми та скрипти, які можна автоматизувати за допомогою Task Scheduler, вбудованого в Windows. Таким чином, можна автоматизувати всі процеси без необхідності придбання дорогих готових рішень.

Для прикладу візьмемо існуючі інструменти, написані на PowerShell, що використовують можливості пакетного менеджера Winget. Winget дозволяє автоматизувати встановлення програмного забезпечення у великих масштабах та безперервно. Завдяки цьому, такі менеджери, як Winget, набули популярності.

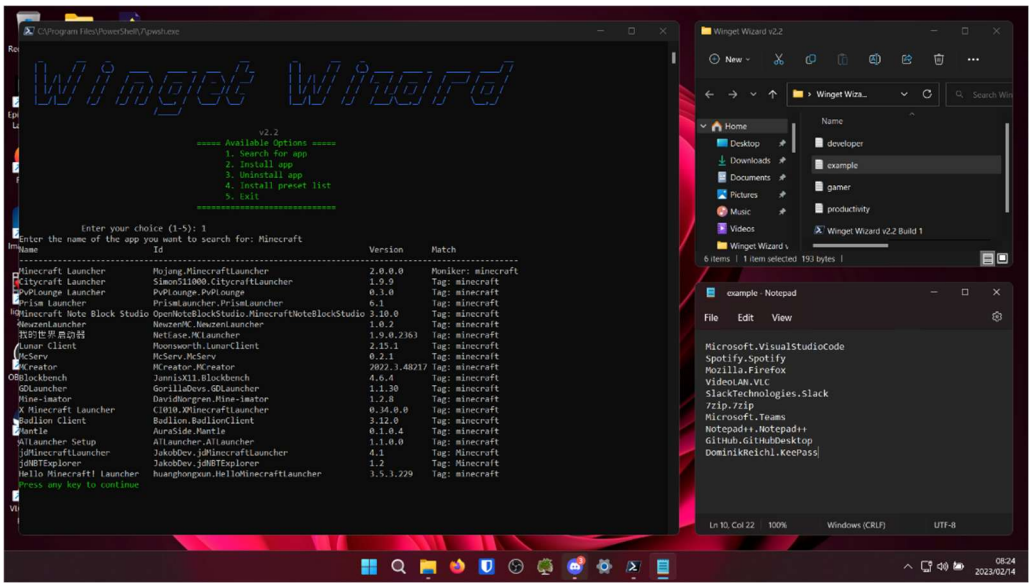

Рисунок 3.4 – Інструмент Winget Wizard на базі можливостей від Winget CLI (GitHub в якості контролю версій ПЗ)

Існують також інструменти, створені на базі Winget, такі як інструмент Chris Titus Tech Windows від Кріса Тітуса, що дозволяє користувачам, які не бажають працювати з кодом, встановлювати різні версії програм, просто обираючи потрібні опції (див. рис. 3.5).

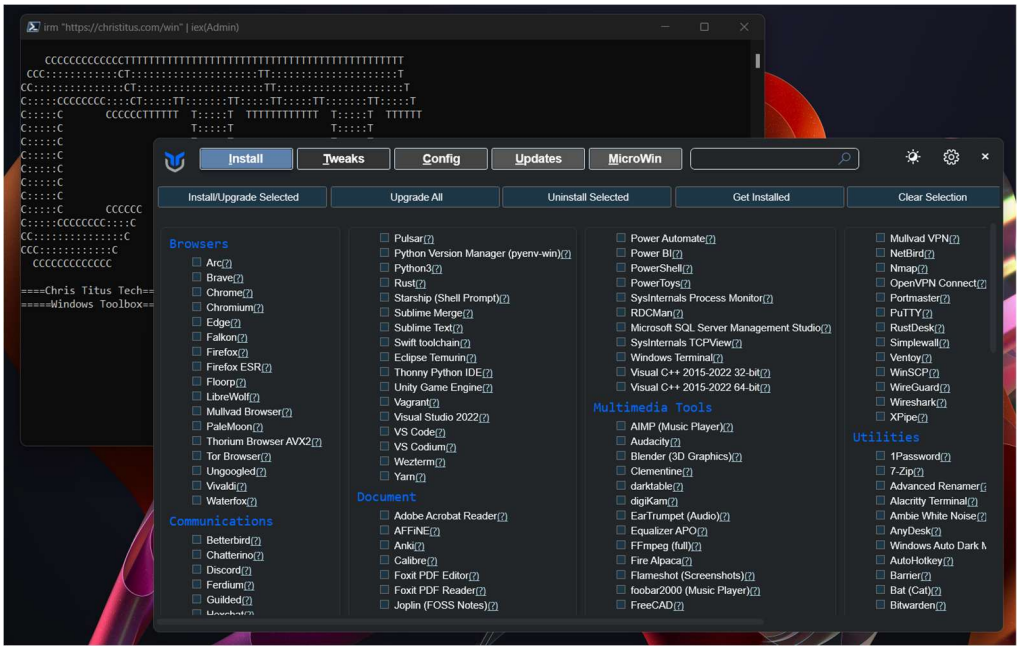

Рисунок 3.5 – Інструмент Chris Titus Tech's Windows Utility на базі можливостей від Winget CLI (GitHub в якості контролю версій ПЗ)

Безпека та відповідність. Останнім, але важливим компонентом архітектури розподіленої системи обслуговування кінцевих, які використовуються віддаленими працівниками, є рівень безпеки та відповідності. Такий компонент включає різноманітні механізми автентифікації, авторизації та шифрування, що гарантують доступ до даних і пристроїв лише авторизованим користувачам та захист від несанкціонованого доступу.

Одним із методів забезпечення безпеки є шифрування каналів зв'язку між кінцевими пристроями та центральним сервером. Таке рішення дозволяє передавати конфіденційні дані, уникаючи ризику їх перехоплення. До того ж, для посилення захисту можуть бути застосовані інструменти контролю доступу до корпоративних акаунтів, як Google Workspace або Microsoft 365, що забезпечують обмежений доступ до ресурсів компанії, що корисно в ситуаціях, коли доступ до корпоративних акаунтів є необхідним для роботи, але повний контроль за ними залишається за компанією (див. рис. 3.6) [38].

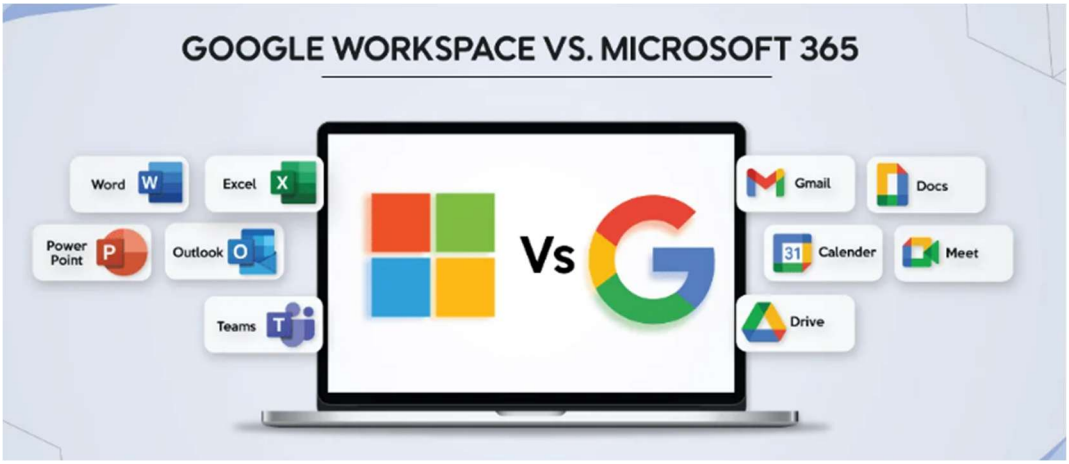

Рисунок 3.6 – Хмарні рішення Google Workspace та Microsoft 365 (для забезпечення безпеки та відповідності)

Сучасні рішення включають також двофакторну автентифікацію та використання апаратних ключів безпеки, наприклад, таких як YubiKey – тип пристроїв, який дозволяє здійснювати автентифікацію користувача на рівні апаратного захисту, забезпечуючи надійний доступ до корпоративних даних. Апаратний ключ захищений від злому завдяки використанню шифрування

високого рівня, і лише власник ключа може підтвердити свою особу для отримання доступу до корпоративних ресурсів. Наприклад, якщо працівник працює з конфіденційними даними або обліковими записами, YubiKey може забезпечити більш надійний захист, ніж традиційні методи авторизації (див. рис. 3.7) [39].

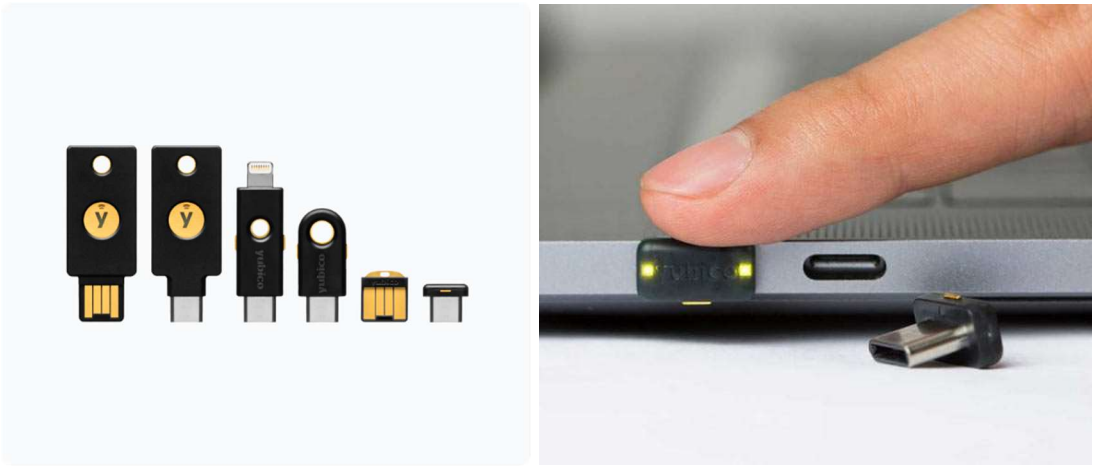

Рисунок 3.7 – Апаратний ключ автентифікації YubiKey

Іншим прикладом апаратного захисту є використання технології довіреної платформи (Trusted Platform Module, TPM), яка надає рівень апаратного шифрування для захисту даних у випадках втрати або крадіжки пристрою. TPM дозволяє здійснювати апаратне шифрування для жорстких дисків, як у рішенні BitLocker, що запобігає доступу зловмисників до даних на пристрої навіть у разі фізичного доступу до нього. TPM, як і BitLocker, інтегрується в систему безпеки, дозволяючи надійно зберігати дані у випадку, якщо пристрій потрапляє до сторонніх осіб (див. рис. 2.23) [23].

Використання таких засобів безпеки забезпечує високий рівень захисту кінцевих пристроїв у розподіленій системі обслуговування, що дозволить віддаленим працівникам працювати з чутливою інформацією без ризику компрометації даних, а IT-команді в компанії – безпечно управляти пристроями, підтримуючи їхній захист як від втрати, так і від стороннього втручання.

Робочий процес в архітектурі автоматизованої розподіленої системи обслуговування кінцевих пристроїв, які використовуються віддаленими працівниками, може виглядати наступним чином:

1. Реєстрація пристроїв: кінцеві пристрої реєструються на центральному сервері через спеціальні агентські програми, звані «агенти кінцевих точок» або «Endpoint Agents». Ці агенти встановлюються на пристроях і постійно працюють у фоновому режимі, що забезпечує їхній зв’язок із сервером та контроль з боку адміністраторів [14, 23, 34].

2. Розподіл завдань від сервера: Сервер керування передає завдання агента кінцевих точок, що можуть включати розгортання нового програмного забезпечення, виконання оновлень чи зміни конфігурацій. Таким чином, всі пристрої отримують необхідні для роботи компоненти без прямого втручання користувачів, що знижує ризик людської помилки [23, 34].

3. Повідомлення про стан завдань: «Агенти» кінцевих точок (Endpoint Agents) регулярно передають дані про статус виконання завдань, що включають успішне або неуспішне виконання операцій, а також виявлення невідповідностей політикам безпеки. Наприклад, якщо встановлення нової програми було завершено успішно, агент передасть відповідне повідомлення, або ж сигналізуватиме про помилку, якщо завдання виконати не вдалося [11, 23, 34].

4. Управління мережею через централізовані панелі: ІТ-команди керують мережею пристроїв за допомогою інформаційних панелей. Тут вони можуть контролювати поточний стан пристроїв, автоматизувати обслуговування та оперативно виявляти проблеми. Панелі дозволяють не тільки відстежувати статус виконання завдань, а й виконувати негайні дії для вирішення проблем [14, 17, 23, 34].

Такий процес забезпечує високий рівень автоматизації і безперервного обслуговування пристроїв, запобігаючи виникненню проблем через застаріле ПЗ чи неправильні налаштування, що особливо корисно для підтримки пристроїв, що перебувають у віддалених локаціях, адже унеможливлює проблеми, які могли б виникнути при обслуговуванні вручну. Наприклад, використання інструментів DevOps, таких як Ansible (не підходить для моніторнгу та автоматизації технічного обслуговування ноутбуків, ПК та мобільних пристроїв), дозволяє ще більше

автоматизувати управління і розгортання змін, що знижує потребу в ручному виконанні рутинних команд і економить час.

Запасний вихід – ручний віддалений доступ. Водночас, для віддаленого керування без необхідності інсталяції складного додаткового ПЗ можна використовувати інструменти типу TeamViewer або AnyDesk. Ці програми також можуть запускатися у фоновому режимі й надавати доступ адміністратору без вимоги від працівника постійно активувати сеанс, що зручно для технічного обслуговування. Однак, на відміну від автоматизованих агентів, у цьому випадку працівник бачить дії адміністратора, що корисно для разового віддаленого керування та підтримки) (див. рис. 2.31).

Microsoft Remote Desktop Connection (більше підходить для локальних рішень, для глобальних краще Windows Remote Desktop Manager) – це рішення для віддаленого доступу, яке має функціонал, схожий на TeamViewer або AnyDesk, але з деякими відмінностями. Основна особливість полягає в тому, що Remote Desktop Connection працює більш непомітно для користувача. Після налаштування відповідних параметрів доступу – локального або зовнішнього – адміністратор може увійти до пристрою працівника і виконувати необхідні дії, а сам працівник цього не побачить, адже доступ відбувається у фоновому режим, що дозволяє IT-командам ефективно керувати налаштуваннями системи без прямого втручання в робочий процес (див. рис. 2.31, 2.32) [32].

Таке рішення є ручним інструментом для підтримки, коли адміністратору потрібно безпосередньо підключитися до пристрою: у TeamViewer чи AnyDesk сесія віддаленого доступу, як правило, помітна для працівника, що може вимагати підтвердження з його боку. У випадку Remote Desktop Connection доступ залишається непомітним, що дозволяє працювати з налаштуваннями системи без додаткового залучення працівника (див. рис. 3.8).

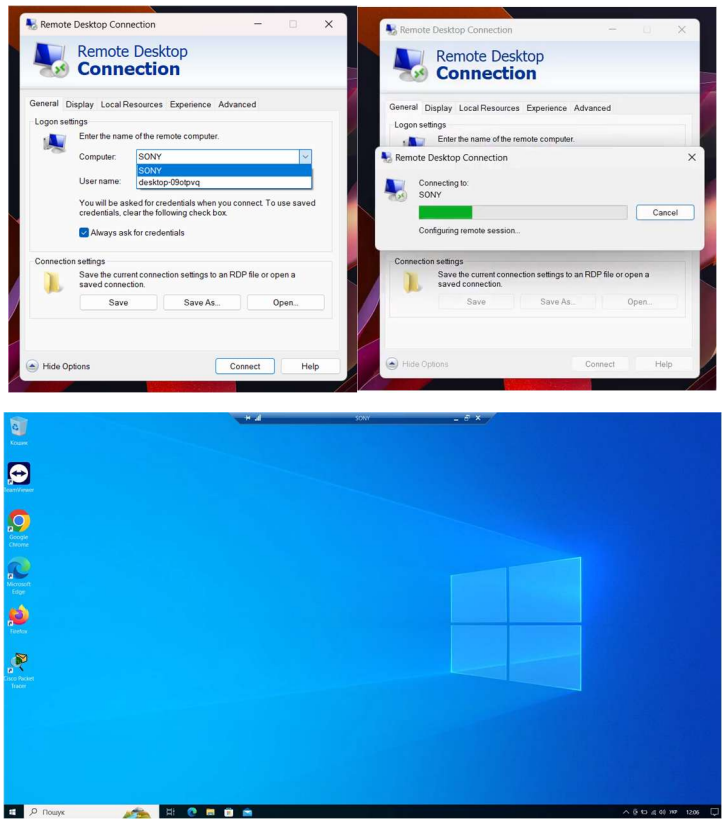

Рисунок 3.8 – Інтерфейс Remote Desktop Connection

Однак, для масштабного обслуговування розподіленої системи потрібно автоматизувати процеси, щоб знизити частоту ручних підключень. Remote Desktop Connection все ще передбачає підключення вручну, тому не зовсім підходить для автоматизації обслуговування великої кількості пристроїв, які використовуються віддаленими працівниками, і взагалі підходить для локальних рішень та націлений лише на одний пристрій У таких випадках важливо використовувати централізовані автоматизовані рішення, які можуть працювати без ручного втручання.

3.2 Алгоритми і сценарії автоматизації обслуговування кінцевих пристроїв

Автоматизація обслуговування кінцевих пристроїв, які використовуються віддаленими працівниками, значно підвищує ефективність завдяки використанню алгоритмів і сценаріїв, які заздалегідь визначають порядок дій. Такі алгоритми та скрипти виконують типові задачі, як розгортання програм, оновлення системи, внесення змін у конфігурації тощо, у фоновому режимі, зберігаючи час та зусилля.

Основою для таких процесів є написання сценаріїв, що забезпечують виконання завдань без залучення працівника. Наприклад, у разі відсутності параметра «тихого режиму» (Silent Mode) або інших параметрів для автоматичної роботи, програми можуть запускати інтерфейс інсталяції безпосередньо на екрані користувача, що створить незручності. Саме для таких випадків потрібні сценарії із налаштуванням потрібних параметрів, що дозволяють виконувати завдання тихо, у режимі фонової роботи .

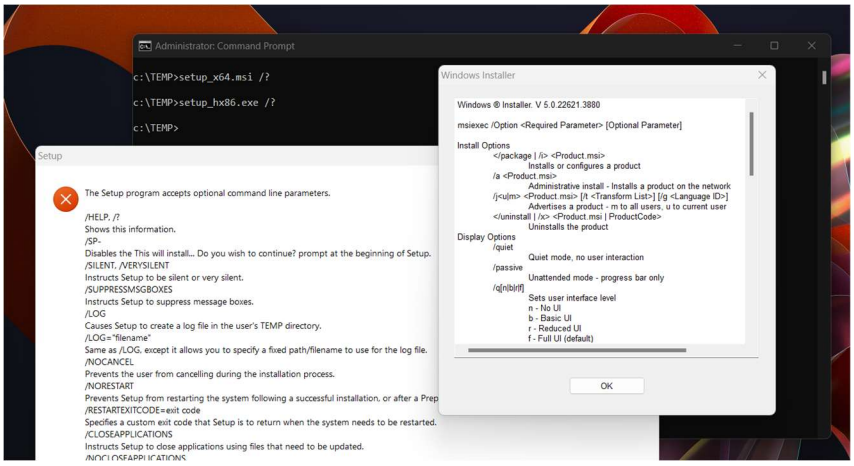

Рисунок 3.9 – Наявність у інсталлерів параметрів для швидкої непомітної розгортки ПЗ

Автоматизовані сценарії особливо важливі для виконання рутинних завдань або масштабованих робіт. Наприклад, якщо адміністратору потрібно оновити

конфігурацію, налаштувати десятки програм або провести однотипні дії на безлічі пристроїв, використання сценарію значно пришвидшить процес, звільнивши час для важливіших завдань.

Для ОС Windows такими інструментами є PowerShell, який дозволить автоматизувати процеси на рівні команд операційної системи, що, своєю чергою, дозволяє налаштовувати конфігурації, розгортати ПЗ та інші системні операції. У Linux і macOS подібні задачі можна виконувати через командну оболонку Bash. Ці інструменти підтримують цикли і конструкції, характерні для мов програмування, що дозволяє виконувати послідовності завдань [28].

Для підвищення продуктивності такі сценарії виконуються у неробочий час, коли пристрій працівника підключений до мережі та перебуває у стані онлайн, що дозволяє запускати процеси непомітно для працівника, мінімізуючи втручання у його робочий день і запобігаючи дискомфорту. До того ж, автоматизація значно розвантажує ІТ-команди, даючи їм змогу зосередитися на нових завданнях, перевірці інших систем чи розробці додаткових сценаріїв, поки процес обслуговування виконується самостійно, без потреби у ручному контролі.

Крім запуску скриптів та сценаріїв вручну, можна налаштувати їх автоматичне виконання за розкладом, що дозволить системі автоматично виконувати певні задачі, наприклад, щотижнево перевіряти стан встановлених програм. Замість того, щоб лише оновлювати чи встановлювати нове ПЗ, ці сценарії можуть перевіряти, чи програма є актуальною, чи використовується працівником, або ж чи відсутня на пристрої.

Сценарії, налаштовані на виконання по розкладу, будуть автоматично здійснювати необхідні оновлення та перевірки без участі ІТ-команди. Наприклад, щопонеділка скрипт може перевіряти наявність конкретного додатку. Якщо він відсутній — встановлює його, а якщо додаток вже присутній — перевіряє, чи оновлений він до останньої версії. Завдяки такій автоматизації зникає потреба постійно пам’ятати про необхідність перевірок або запуску процесів вручну.

В ОС Windows для реалізації таких регулярних завдань існує спеціальна утиліта Windows Task Scheduler. Вона дозволяє налаштувати періодичне виконання

скриптів і сценаріїв, забезпечуючи безперебійну та регулярну підтримку необхідного стану програмного забезпечення на пристроях, які використовуються віддаленими працівниками.

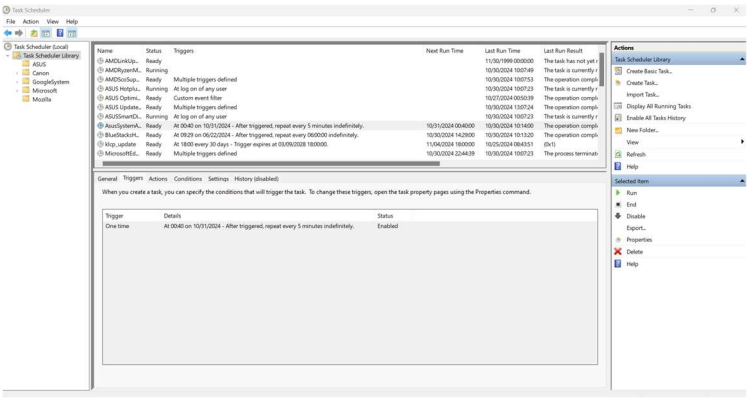

Рисунок 3.10 – Використання Task Scheduler як заплановане виконання сценаріїв (скриптів) автоматизації

Приклад скрипта на PowerShell для автоматичного встановлення програми, що перевіряє її наявність, а в разі відсутності — інсталює програму (див. рис. 3.11) [40].

```powershell
#Налаштування Windows Remote Management (WinRM) (для запуску команд на віддалених ПК)

winrm quickconfig -q
winrm set winrm/config/client '@{TrustedHosts="*"}'

#Копіювання файлу на віддалений пристрій

$RemoteComputer = "RemotePC"   # Ім'я або IP віддаленого комп'ютера
$InstallerPath = "C:\Installers\software_installer.exe" # Локальний шлях до інсталятора
$RemotePath = "\\$RemoteComputer\C$\Temp" # Шлях на віддаленому ПК
$SoftwareName = "SoftwareName" # Назва ПЗ для перевірки
$Credentials = Get-Credential  # Запит на введення облікових даних

# Перевірка наявності ПЗ на віддаленому ПК

$SoftwareInstalled = Invoke-Command -ComputerName $RemoteComputer -Credential $Credentials -ScriptBlock {
    param ($SoftwareName)
    $path = "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall"
    $software = Get-ChildItem -Path $path | ForEach-Object {
        Get-ItemProperty -Path $_.PSPath
    } | Where-Object { $_.DisplayName -like "*$SoftwareName*" }

    return $software -ne $null  # Повертає true, якщо ПЗ знайдено
} -ArgumentList $SoftwareName

if ($SoftwareInstalled) {
    Write-Output "Програмне забезпечення $SoftwareName вже встановлено на $RemoteComputer."
} else {
    Write-Output "Програмне забезпечення $SoftwareName не знайдено на $RemoteComputer. Починаємо копіювання та встановлення."

    # Копіюємо інсталятор на віддалений пристрій

    Copy-Item -Path $InstallerPath -Destination $RemotePath

    # Запускаємо інсталяцію на віддаленому пристрої

    Invoke-Command -ComputerName $RemoteComputer -Credential $Credentials -ScriptBlock {
        Start-Process -FilePath "C:\Temp\software_installer.exe" -ArgumentList '/silent' -Wait
    }

    # Після встановлення видаляємо інсталятор

    Invoke-Command -ComputerName $RemoteComputer -ScriptBlock {
        Remove-Item -Path "C:\Temp\software_installer.exe"
    } -Credential $Credentials

    Write-Output "Програмне забезпечення $SoftwareName успішно встановлено на $RemoteComputer."
}
```

Рисунок 3.11 – Сценарії (скрипт) автоматизації з логікою перевірки та розповсюдження ПЗ через PowerShell

Алгоритм розгортання програмного забезпечення на віддалених пристроях можна описати більш детально, розглядаючи його кроки від перевірки встановленого ПЗ (див. рис. 3.11):

1. Перевірка наявності ПЗ на кожному пристрої: спочатку алгоритм визначає, чи інстальоване на пристрої потрібне ПЗ, що дозволить уникнути повторної інсталяції програми тієї ж самої версії, зберігаючи ресурси системи.

2. Перевірка версії ПЗ: якщо програма встановлена, наступний крок — перевірка її актуальної версії. Алгоритм з'ясовує, чи відповідає версія програми останній доступній, що забезпечує актуальність ПЗ на всіх пристроях.

3. Оновлення або інсталяція ПЗ: у випадку застарілої версії або відсутності ПЗ на пристрої, автоматично запускається сценарій інсталяції або оновлення, і це може здійснюватись, наприклад, через PowerShell чи CMD для Windows або Bash для Linux та macOS. Навіть, якщо програма відсутня на пристрої, алгоритм її встановить з нуля.

4. Перевірка успішності установки: після завершення процесу інсталяції/оновлення виконується перевірка, щоб переконатися, що операція пройшла успішно. Якщо ПЗ інстальовано без помилок, скрипт надсилає звіт на централізований сервер.

5. Фіксація результатів на сервері: результати кожного розгортання фіксуються в базі даних: зазначається час, версія ПЗ, статус виконання та можливі помилки – це забезпечує ІТ-команду усією необхідною інформацією для подальшого моніторингу та аналізу.

До попереднього алгоритму розгортання програмного забезпечення можна додати детальні алгоритми для керування оновленнями, конфігурацією, відповідністю вимогам безпеки та моніторингом справності пристроїв – ці сценарії забезпечують повну автоматизацію обслуговування та мінімальне втручання адміністратора, фокусуючи зусилля на віддаленому моніторингу та підтримці продуктивності.

Алгоритм керування оновленнями. Регулярні оновлення ПЗ та виправлення помилок є критичними для стабільності системи. Алгоритм запланує періодичну

перевірку оновлень на основі політики організації, перевіряючи, чи наявні нові виправлення або оновлення системи. За необхідності, він завантажує та інсталює оновлення у непіковий час, коли комп’ютер не використовується на повну потужність, щоб мінімізувати ризик збоїв. У разі невдалого оновлення автоматично застосовується механізм відкату, що повертає систему до попередньої стабільної версії.

Алгоритм конфігурації та відповідності. Даний алгоритм автоматично підтримує відповідність пристрою політикам організації, таким як вимоги до паролів чи налаштування VPN. Сценарії для перевірки відповідності періодично запускаються на кожному пристрої та автоматично усувають будь-які невідповідності. Цей підхід аналогічний алгоритму розгортання ПЗ, але з фокусом на підтримці конфігураційної відповідності вимогам організації [8].

Алгоритм моніторингу справності пристроїв та автоматичного самовідновлення. Для забезпечення стабільної роботи пристроїв необхідний постійний моніторинг параметрів, таких як навантаження на процесор, використання оперативної та зовнішньої пам’яті. Якщо будь-який з цих показників перевищує задані порогові значення, алгоритм автоматично очищує тимчасові файли, завершує непотрібні фонові процеси або перезавантажує пристрій. Цей сценарій працює автономно, а інформація про дії та результати передається на сервер моніторингу, що спрощує контроль над загальною стабільністю системи [23].

Налаштування середовища Windows для написання сценаріїв. Для створення власних алгоритмів, не прив'язуючись до готових рішень, необхідно розглянути пакетний менеджер Winget з широкими можливостями. Для скриптингу, тобто написання сценаріїв автоматизації в Windows, найпростішим є командний рядок завдяки зрозумілому синтаксису, що дозволяє реалізувати потрібні завдання.

Для цього на всіх комп'ютерах з Windows, починаючи з Windows 7, підтримується Winget. Але для його коректної роботи потрібно оновити Windows Installer, де міститься цей пакетний менеджер. Оновлення можна виконати

командою в PowerShell (більш просунутий командний рядок) або через звичайний командний рядок.

За замовчуванням можна встановити PowerShell 7 з офіційного сайту, завантаживши MSI-пакет. Сценарії можна запускати через Task Scheduler для постійної перевірки. Встановлення PowerShell 7 (див. рис. 3.12) з MSI-пакета відобразиться у самому PowerShell, надаючи відповідну інформацію, але для роботи пакетного менеджера Winget достатньо і PowerShell 5 версії, що є на всіх ПК, починаючи з Windows 10.

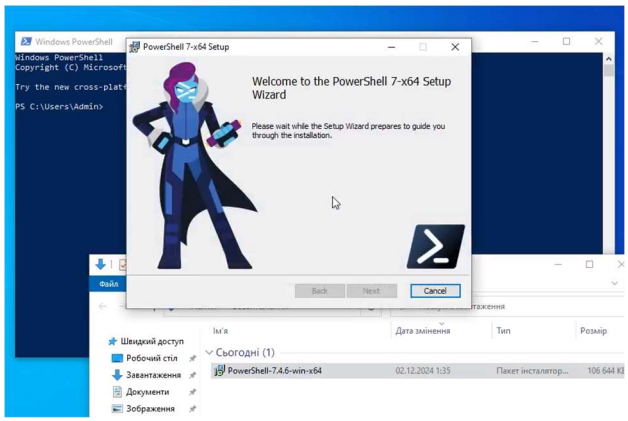

Рисунок 3.12 – Встановлення PowerShell 7 (ручним методом)

Якщо встановити Winget командою не вдається (див. рис.3.13), найпростіший спосіб (див. рис. 3.14) – зайти в Microsoft Store та в розділі оновлень ПЗ знайти та оновити Windows Installer. Після цього в PowerShell сьомої версії можна буде працювати з Winget.

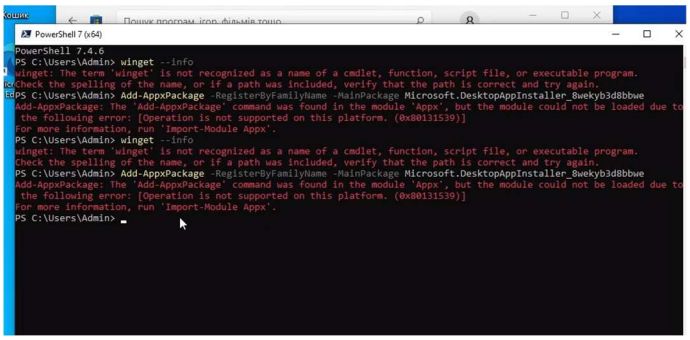

Рисунок 3.13 – Після введення команди для оновлення Windows Installer може виникнути помилка

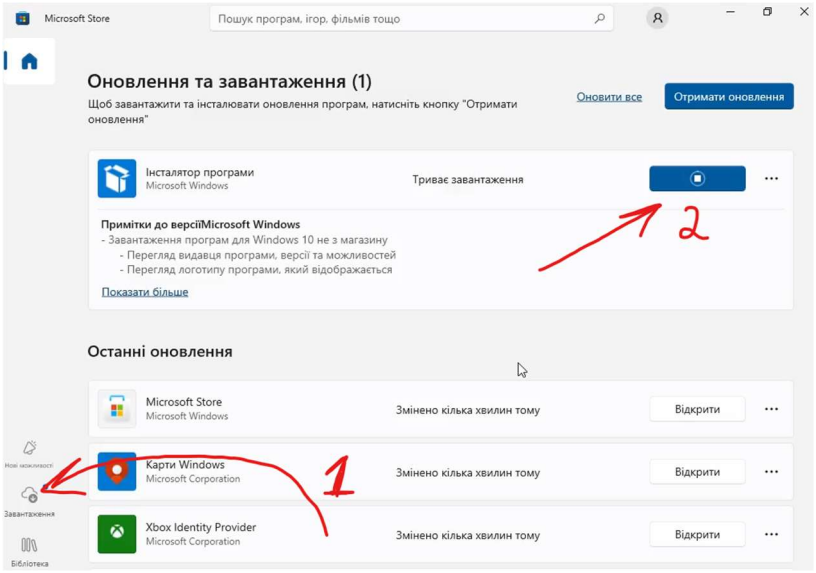

Рисунок 3.14 – Оновлення Windows Installer через Microsoft Store (для отримання Winget)

Після цих кроків у нас має запрацювати пакетний менеджер Winget (див. рис. 3.15), після чого можна переходити до створення скриптів автоматизації.

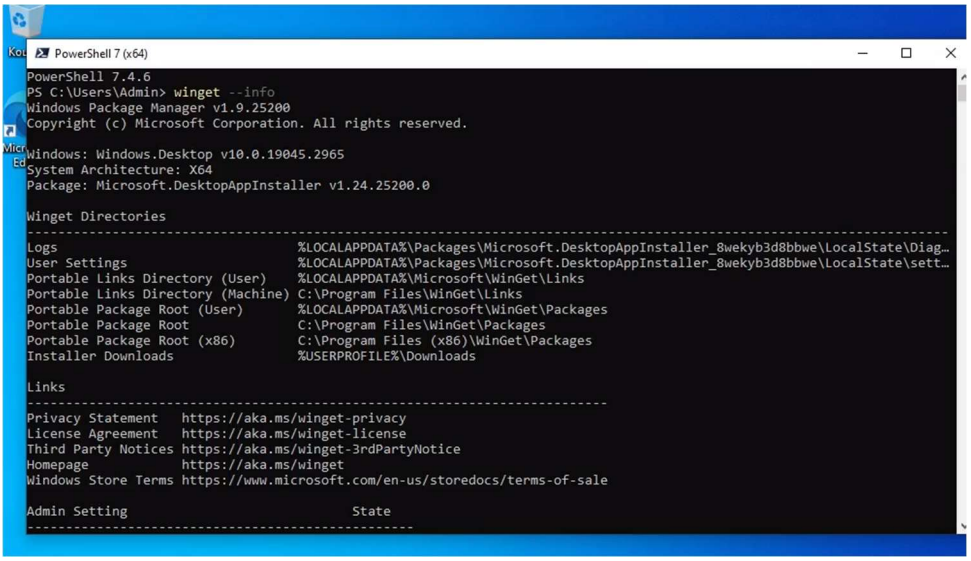

Рисунок 3.15 – Перевірка валідності оновлення та отримання пакетного менеджера Winget

Далі розглянемо кілька базових сценаріїв, які демонструють використання пакетного менеджера Winget для роботи з ПЗ, що стануть основою для більш

масштабних автоматизованих сценаріїв, написаних у командному середовищі Windows – CMD.

```
@echo off
setlocal enabledelayedexpansion

:: Specify the Name of the Software and the Required Version
set SOFTWARE=Google Chrome
set REQUIRED_VERSION=117.0.0

:: Check the Installed Software Version
for /f "tokens=*" %%A in ('winget list --name "!SOFTWARE!" 2^>nul ^| findstr /i
"!SOFTWARE!"') do (
    for /f "tokens=3 delims= " %%B in ("%%A") do set INSTALLED_VERSION=%%B
)

if defined INSTALLED_VERSION (
    echo Installed version: !INSTALLED_VERSION!
    if "!INSTALLED_VERSION!" GEQ "!REQUIRED_VERSION!" (
        echo !SOFTWARE! Version is Up-to-Date.
        exit /b 0
    )
)

echo !SOFTWARE! is Not Installed or Requires an Update.

:: Silent software installation
winget install --id Google.Chrome -e --silent

if %errorlevel% EQU 0 (
    echo Installation Completed Successfully.
) else (
    echo Installation Failed With Error Code %errorlevel%.
)
```

Рисунок 3.16 – Сценарій перевірки наявності ПЗ потрібної версії (у разі відсутності – проводиться «тиха» інсталяція)

Сценарій (див. рис. 3.16) є прикладом командного файлу для перевірки наявності встановленого ПЗ визначеної версії, з можливістю автоматичної «тихої» інсталяції у разі невідповідності версії або відсутності ПЗ. Після активації командний файл використовує змінну "SOFTWARE" для визначення назви ПЗ (у цьому випадку – Google Chrome) та змінну "REQUIRED_VERSION" для встановлення потрібної версії (у цьому випадку саме 117.0.0).

На першому етапі сценарій перевіряє встановлену версію ПЗ за допомогою утиліти Winget, яка виконує пошук інсталяційного пакету, що відповідає назві «SOFTWARE». Результати обробляються через вкладені цикли "for", які

отримують версію зі списку встановлених програм та присвоюють її змінній "INSTALLED_VERSION".

Далі реалізується перевірка: якщо змінна "INSTALLED_VERSION" існує, то перевіряється відповідність її значення з необхідною версією. Якщо встановлена версія дорівнює або перевищує потрібну, на екран виводиться повідомлення про актуальність версії, і виконання сценарію завершується успішно. У протилежному випадку або за відсутності ПЗ виводиться повідомлення про необхідність інсталяції або оновлення.

На наступному етапі виконується автоматична «тиха» інсталяція ПЗ за допомогою команди winget install із параметрами "--silent", що забезпечує встановлення без втручання користувача. Результат інсталяції оцінюється за значенням змінної "%errorlevel%": у разі успішної інсталяції користувач отримує повідомлення про її завершення, інакше виводиться повідомлення про помилку із зазначенням відповідного коду.

```
@echo off
setlocal enabledelayedexpansion

:: Specify the Name of the Software and the Required Version
set SOFTWARE=Google Chrome
set REQUIRED_VERSION=117.0.0

:: Check the Installed Software Version
for /f "tokens=*" %%A in ('winget list --name "!SOFTWARE!" 2^>nul ^| findstr /i
"!SOFTWARE!"') do (
    for /f "tokens=3 delims= " %%B in ("%%A") do set INSTALLED_VERSION=%%B
)

if defined INSTALLED_VERSION (
    echo Installed Version: !INSTALLED_VERSION!
    if "!INSTALLED_VERSION!" GEQ "!REQUIRED_VERSION!" (
        echo !SOFTWARE! is Up-to-Date. No Update is Required.
        exit /b 0
    )
)

echo Updating !SOFTWARE! to Version !REQUIRED_VERSION!.
:: Silent Software Update
winget upgrade --id Google.Chrome -e --silent

if %errorlevel% EQU 0 (
    echo Update Completed Successfully.
) else (
    echo Update Failed With Error Code %errorlevel%.
)
```

Рисунок 3.17 – Сценарій оновлення ПЗ

(у разі актуальності ПЗ– оновлення не проводиться)

Наступний сценарій (див. рис.3.17) автоматизує процес перевірки версії програмного забезпечення (ПЗ) та його оновлення за допомогою пакетного менеджера Winget. Цей сценарій використовується для забезпечення актуальності визначеного ПЗ, виключаючи необхідність ручного втручання. У цьому прикладі розглядається оновлення Google Chrome до версії 117.0.0.

Спочатку визначається назва ПЗ та його необхідна версія шляхом встановлення значень змінних "SOFTWARE" і "REQUIRED_VERSION". Далі, за допомогою команди "winget list", здійснюється пошук встановленого ПЗ. Використовується цикл "for", який обробляє вихідний список програм, отримуючи встановлену версію і присвоюючи її змінній "INSTALLED_VERSION".

Після цього виконується перевірка: якщо ПЗ встановлено (змінна "INSTALLED_VERSION" визначена), сценарій порівнює її значення з необхідною версією "REQUIRED_VERSION". Якщо встановлена версія дорівнює або перевищує необхідну, виводиться повідомлення про актуальність ПЗ, і сценарій завершується без подальших дій.

Якщо ж версія ПЗ застаріла або його взагалі не встановлено, сценарій ініціює процес «тихого» оновлення за допомогою команди winget upgrade з параметрами --silent, які забезпечують виконання операції без втручання користувача.

Результат оновлення оцінюється через перевірку змінної "%errorlevel%". Якщо команда виконується успішно, користувач отримує повідомлення про успішне завершення оновлення. У разі виникнення помилки виводиться відповідне повідомлення із зазначенням коду помилки, що дозволяє ідентифікувати причину збою. Сценарій забезпечує як перевірку, так і автоматизацію оновлення ПЗ, значно полегшуючи адміністративні завдання.

```
@echo off
setlocal enabledelayedexpansion

:: Specify the Path to the Directory to be Cleaned
set DIR_TO_CLEAN=C:\Temp

:: Check Whether the Directory Exists
if not exist "!DIR_TO_CLEAN!" (
    echo Directory "!DIR_TO_CLEAN!" Does Not Exist.
    exit /b 1
)

echo Cleaning Directory: "!DIR_TO_CLEAN!"

:: Delete All Files
del /q "!DIR_TO_CLEAN!\*.*"

:: Deleting All Subdirectories and Their Contents
for /d %%D in ("!DIR_TO_CLEAN!\*") do (
    rd /s /q "%%D"
)

echo Directory Cleaned Successfully.
exit /b 0
```

Рисунок 3.18 – Сценарій оновлення ПЗ

(у разі актуальності ПЗ– оновлення не проводиться)

Сценарій (див. рис. 3.18) автоматизує процес очищення вмісту вказаної директорії, включаючи всі файли та підкаталоги. Це корисний інструмент для управління тимчасовими файлами, звільнення дискового простору та підтримки чистоти файлової системи. У даному прикладі очищується директорія "C:\Temp".

На початку сценарій визначає шлях до директорії, яка потребує очищення, шляхом встановлення змінної "DIR_TO_CLEAN". Потім виконується перевірка наявності цієї директорії за допомогою умови "if not exist". Якщо директорія не існує, сценарій виводить повідомлення про це і завершується з кодом помилки "1", що свідчить про невиконання операції.

Якщо директорія існує, виводиться повідомлення про початок очищення. Першим етапом очищення є видалення всіх файлів у кореневій частині директорії. Це реалізується за допомогою команди "del /q", яка працює у тихому режимі (/q), не вимагаючи підтвердження для видалення кожного файлу.

Другим етапом є видалення всіх підкаталогів разом із їх вмістом. Для цього використовується цикл "for /d", який ітерується через всі підкаталоги вказаної

директорії. Кожен підкаталог видаляється разом із вмістом за допомогою команди "rd /s /q", де "/s" забезпечує видалення вмісту підкаталогів, а "/q" – тихий режим виконання.

Після успішного видалення всіх файлів і підкаталогів виводиться повідомлення про завершення очищення директорії. Сценарій завершується з кодом "0", що свідчить про успішне виконання операції. Такий сценарій є ефективним інструментом для автоматизації очищення директорій, наприклад, із тимчасовими файлами.

```
@echo off
setlocal enabledelayedexpansion

:: Specify the IP Address or Domain to Test the Connection
set TEST_ADDRESS=8.8.8.8

:: Specify the Name of The Network Adapter
set ADAPTER_NAME="Ethernet"

echo Checking Network Connection...

:: Checking the Availability of the Address
ping -n 1 %TEST_ADDRESS% >nul 2>&1
if %errorlevel% EQU 0 (
    echo Network is Connected.
    exit /b 0
)

echo Network is Not Reachable. Restarting the Network Adapter...

:: Disconnecting the Adapter
netsh interface set interface name=%ADAPTER_NAME% admin=disable
timeout /t 5 >nul

::Enabling the Adapter
netsh interface set interface name=%ADAPTER_NAME% admin=enable
timeout /t 5 >nul

:: Checking the Connection Again
ping -n 1 %TEST_ADDRESS% >nul 2>&1
if %errorlevel% EQU 0 (
    echo Network Reconnected Successfully.
    exit /b 0
) else (
    echo Failed to Reconnect to the Network.
    exit /b 1
)
```

Рисунок 3.19 – Сценарій перевірки з’єднання з мережею (у разі відсутності – перезавантажує мережевий адаптер)

Сценарій (див. рис.3.19) автоматизує перевірку доступності мережевого з'єднання шляхом надсилання запиту до певної IP-адреси або домену, і, у разі відсутності з'єднання, перезавантажує мережевий адаптер для спроби відновлення підключення, що може бути корисним для діагностики і швидкого виправлення мережевих збоїв.

На початку сценарію задаються два параметри: IP-адреса або домен для перевірки доступності, що зберігається у змінній "TEST_ADDRESS " (в прикладі це "8.8.8.8", DNS-сервер Google), і назва мережевого адаптера у змінній "ADAPTER_NAME " (в даному випадку – "Ethernet").

Сценарій розпочинає роботу з виведення повідомлення "Checking Network Connection..." і спроби перевірити доступність заданої адреси за допомогою команди "ping -n 1 %TEST_ADDRESS%". Опція -n 1 вказує на надсилання одного запиту. Якщо запит успішний (код повернення %errorlevel% дорівнює 0), сценарій виводить повідомлення "Network is Connected." і завершує роботу.

Якщо адреса недоступна, виводиться повідомлення "Network is Not Reachable. Restarting the Network Adapter...", і виконується перезавантаження мережевого адаптера. Для цього адаптер вимикається командою "netsh interface set interface name=%ADAPTER_NAME% admin=disable", після чого сценарій робить паузу у 5 секунд за допомогою команди "timeout /t 5 >nul". Потім адаптер знову вмикається командою "netsh interface set interface name=%ADAPTER_NAME% admin=enable", і знову виконується пауза у 5 секунд.

Після перезавантаження мережевого адаптера виконується повторна перевірка доступності адреси тією ж командою "ping -n 1 %TEST_ADDRESS%". Якщо підключення відновлено, сценарій виводить повідомлення "Network Reconnected Successfully." і завершує роботу. У разі невдачі виводиться повідомлення "Failed to Reconnect to the Network.", і сценарій завершується з кодом помилки "1". Такий сценарій забезпечує ефективну діагностику та базове автоматизоване виправлення проблем з мережею, мінімізуючи необхідність ручного втручання.

Розроблений сценарій реалізації безперервного першочергового обслуговування пристроїв. Розроблений сценарій (див. рис.3.20) (Додаток Б, В) є масштабним і детально продуманим рішенням – представляє собою автоматизовану програму для завантаження та встановлення актуальних версій ПЗ на ПК з урахуванням архітектури системи.

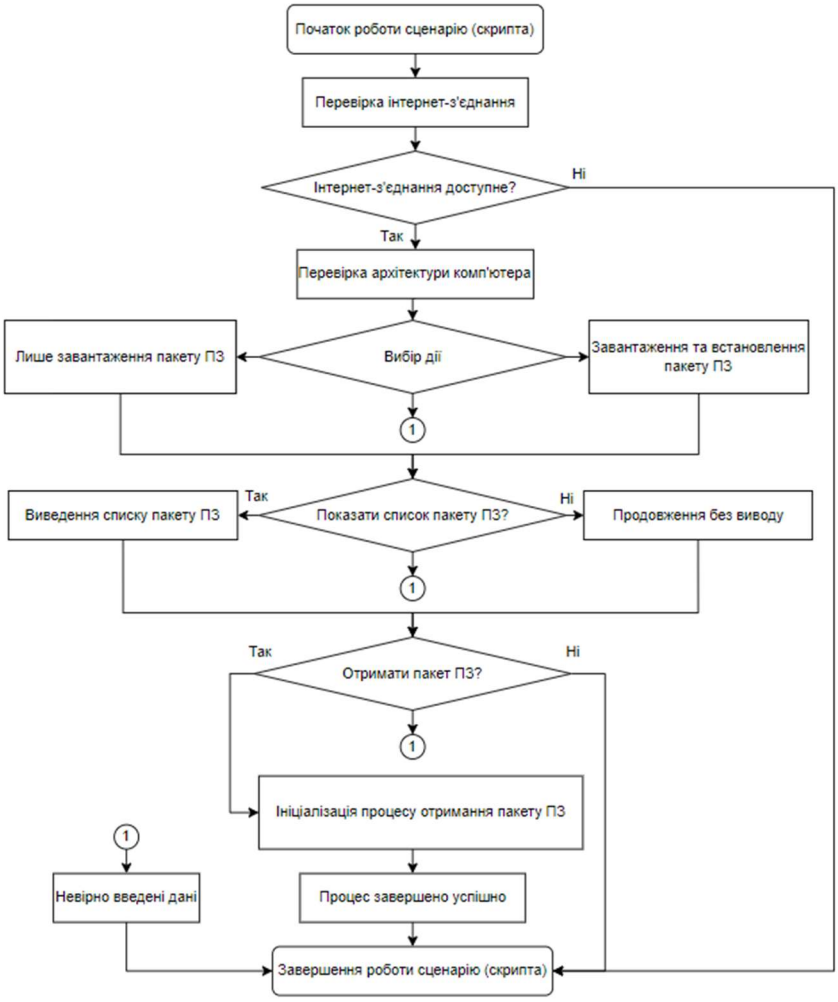

Рисунок 3.20 – Блок-схема роботи сценарію безперервного розгортання ПЗ

Основною метою є створення універсального рішення, яке може працювати на будь-якому комп’ютері, для швидкого розгортання ПЗ, що містить різні пакети ПЗ для забезпечення першочергового обслуговування пристроїв (див. рис.3.21).

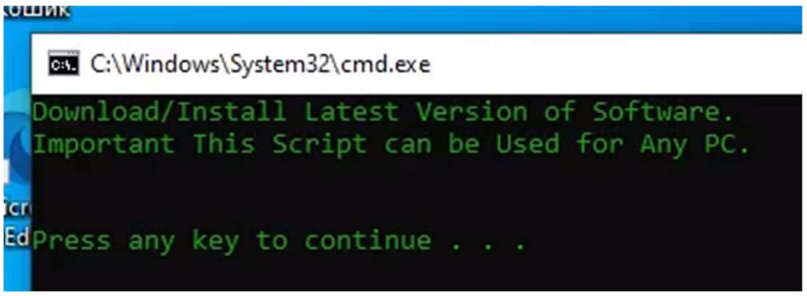

Рисунок 3.21 – Початкова шапка-опис сценарію

На початку сценарію проводиться перевірка доступності інтернет-з'єднання шляхом пінгу IP-адреси Google DNS-сервера. Якщо підключення відсутнє (див. рис. 3.22), скрипт завершує роботу з відповідним повідомленням.

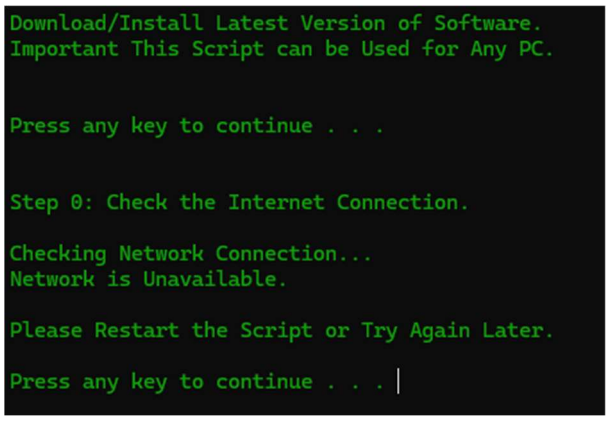

Рисунок 3.22 – Попереднє завершення роботи скрипта, у разі відсутності Інтернет-з’єднання

Якщо ж користувач обере режим виконання (див.рис.3.23): лише завантаження інсталяційних файлів ("Download") або завантаження з подальшою установкою ("Install"). У разі вибору режиму встановлення визначається архітектура системи (32- або 64-бітна) (не виводиться на екран), після чого відповідні параметри налаштовуються для подальшого виконання.

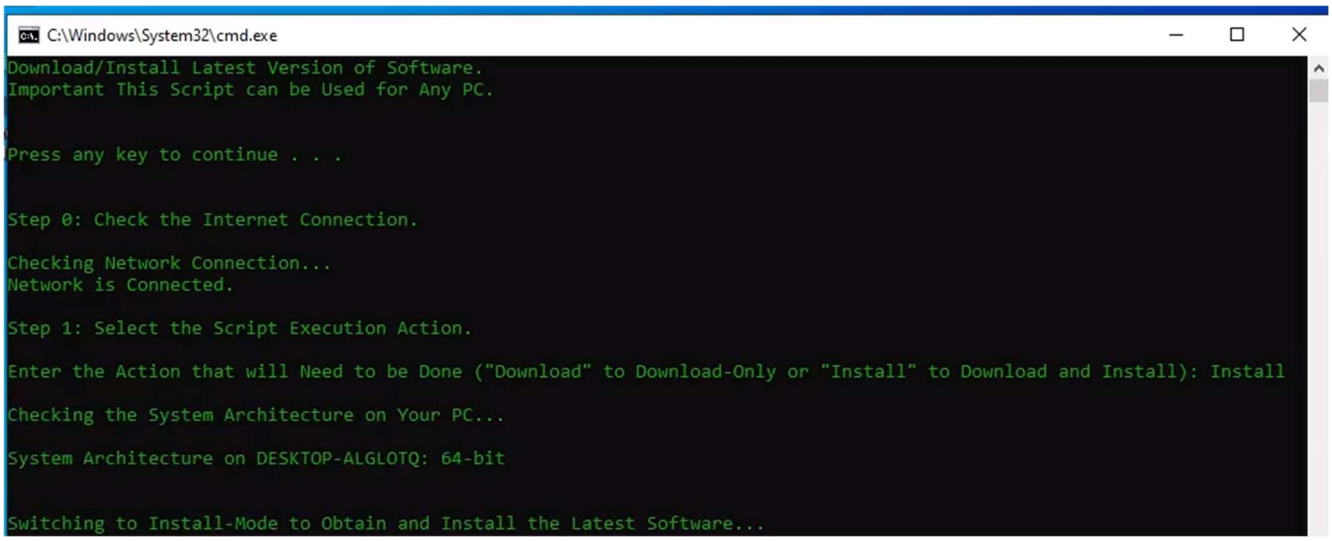

Рисунок 3.23 – Вибір дії сценарію виконання
(лише завантажити пакет ПЗ або завантажити та встановити пакет ПЗ)

Далі, після обрання будь-якої із дій, запитується про те, чи вивести список пакету програм, які будуть завантажені та/або інстальвані (див. рис. 3.24).

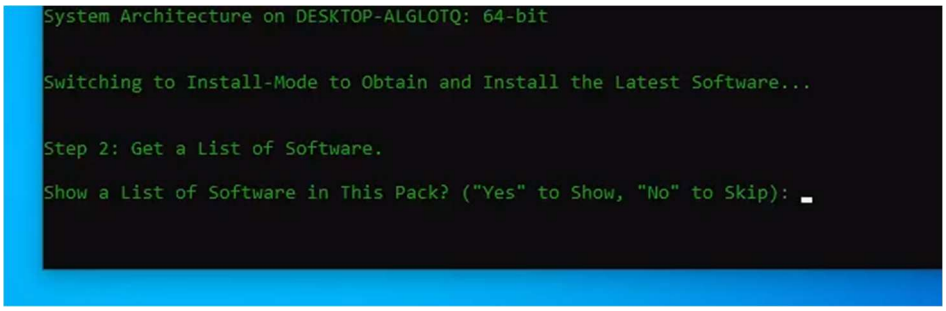

Рисунок 3.24 – Обрано дію «завантажити та встановити пакет ПЗ» та
перехід до іншого кроку – показати пакет ПЗ

Після вибору «показати список програм» нам виводиться весь список пакету ПЗ, яке пропонується завантажити та/або встановити. Загалом, скрипт організований у 15 категорій (див. рис.3.25), кожна з яких виконує окрему функцію для покриття всіх ключових аспектів роботи комп’ютерної системи.

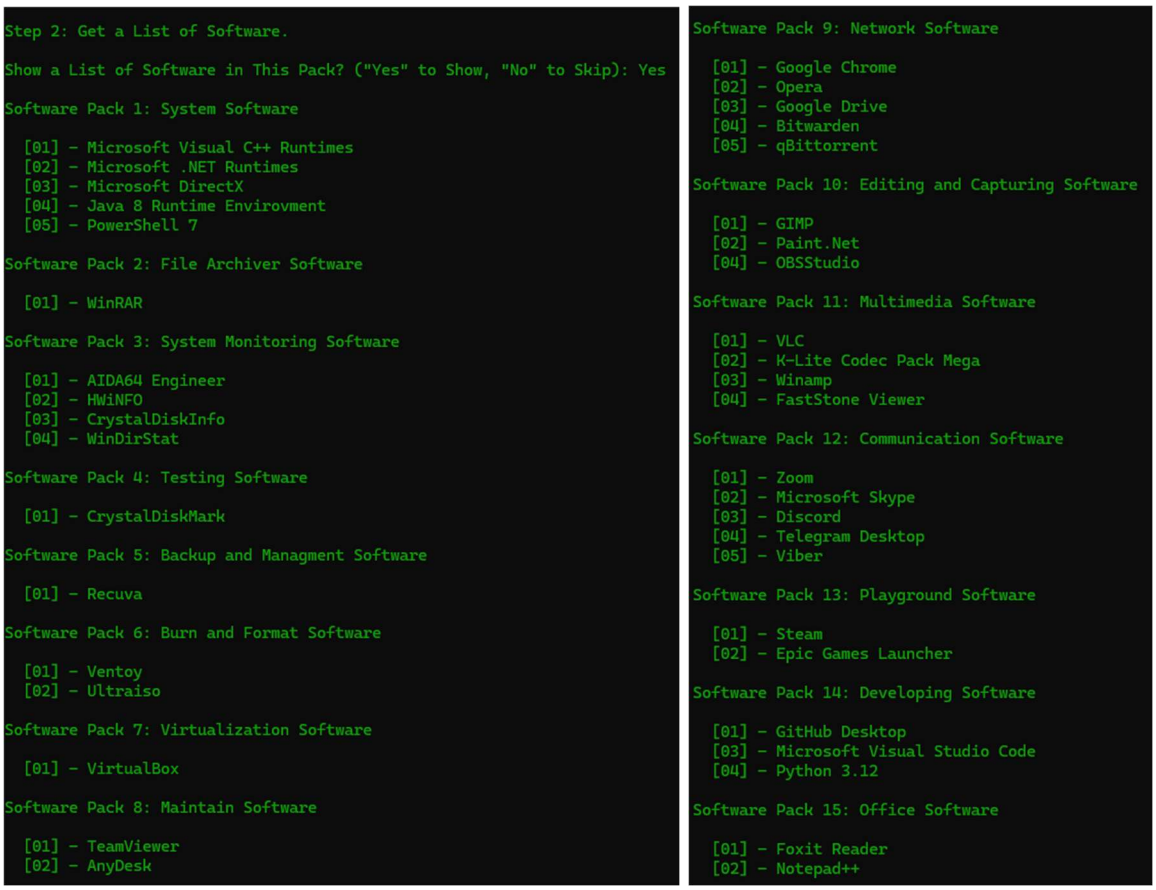

Рисунок 3.25 – Список пакету ПЗ, до якого входить 15 підпакетів, поділених на категорії

"Software Pack 1: System Software" (див. рис.3.25) забезпечує основні потреби системи. У цей пакет входять Microsoft Visual C++ Runtimes, Microsoft .NET Runtimes, Microsoft DirectX, Java 8 Runtime Environment і PowerShell 7. Ці компоненти формують базу для роботи інших програм, забезпечуючи сумісність, підтримку критичних бібліотек і виконання скриптів. Без цих елементів подальші налаштування були б неможливими.

"Software Pack 2: File Archiver Software " (див. рис.3.25) орієнтований на обробку файлів. Він містить WinRAR, популярний інструмент для стиснення та розпакування даних. Наявність архіватора спрощує роботу з великими файлами та дозволяє економити місце на диску.

"Software Pack 3: System Monitoring Software" (див. рис.3.25) включає інструменти для глибокого аналізу системи: AIDA64 Engineer для діагностики обладнання, HWiNFO для моніторингу апаратних компонентів, CrystalDiskInfo для

перевірки стану жорстких дисків і WinDirStat для візуального аналізу використання дискового простору. Цей пакет дозволяє не лише виявляти потенційні проблеми, але й проактивно запобігати їм.

"Software Pack 4: Testing Software " (див. рис.3.25) включає CrystalDiskMark, інструмент для тестування швидкості носіїв даних. Це дозволяє оцінити продуктивність жорстких дисків і флеш-накопичувачів, що особливо важливо при роботі з великими обсягами даних.

"Software Pack 5: Backup and Management Software" (див. рис.3.25) містить Recuva — інструмент для відновлення видалених файлів. Це критично важлива функція для забезпечення збереження даних у випадку помилок чи збоїв.

"Software Pack 6: Burn and Format Software" (див. рис.3.25) включає Ventoy і UltraISO, які дозволяють створювати завантажувальні флешки, працювати з образами дисків і форматувати носії даних. Цей пакет незамінний для оновлення операційної системи та інших завдань, пов’язаних із підготовкою носіїв.

"Software Pack 7: Virtualization Software" (див. рис.3.25) представлений VirtualBox, потужним інструментом для створення віртуальних машин. Це дозволяє тестувати програмне забезпечення чи працювати з іншими операційними системами без необхідності встановлювати їх на основний пристрій.

"Software Pack 8: Maintain Software" (див. рис.3.25) зосереджується на віддаленому доступі. TeamViewer і AnyDesk дозволяють керувати пристроями дистанційно, забезпечуючи швидке реагування на проблеми без фізичної присутності біля комп’ютера.

"Software Pack 9: Network Software" (див. рис.3.25) забезпечує широкий спектр інструментів для роботи з мережею, включаючи браузери (Google Chrome, Opera), хмарне сховище (Google Drive), менеджер паролів (Bitwarden) і торрент-клієнт (qBittorrent). Це необхідно для зручної роботи в інтернеті та управління даними.

"Software Pack 10: Editing and Capturing Software" (див. рис.3.25) містить програми для роботи з графікою та запису екрана, такі як GIMP, Paint.Net і OBS

Studio. Ці інструменти дозволяють створювати та редагувати контент, а також захоплювати відео та транслювати його.

"Software Pack 11: Multimedia Software" (див. рис.3.25) включає VLC для відтворення відео, K-Lite Codec Pack Mega для забезпечення сумісності форматів, Winamp для аудіо та FastStone Viewer для перегляду фотографій. Цей пакет охоплює всі потреби мультимедійного користувача.

"Software Pack 12: Communication Software" (див. рис.3.25) забезпечує спілкування через Zoom, Microsoft Skype, Discord, Telegram Desktop і Viber. Ці програми дозволяють користувачам залишатися на зв’язку та брати участь у відеоконференціях чи текстових чатах.

"Software Pack 13: Playground Software" (див. рис.3.25) орієнтований на розваги та включає Steam і Epic Games Launcher, які є популярними платформами для доступу до ігор.

"Software Pack 14: Developing Software" (див. рис.3.25) пропонує GitHub Desktop для роботи з репозиторіями, Microsoft Visual Studio Code як середовище розробки та Python 3.12 для програмування. Це ідеальний набір для розробників.

"Software Pack 15: Office Software" (див. рис.3.25) містить Foxit Reader для роботи з PDF-документами та Notepad++ для редагування тексту. Ці інструменти є важливими для офісних завдань.

Кожен із пакетів є частиною великої системи, яка автоматизує процеси, зменшує навантаження на користувача та дозволяє швидко налаштувати пристрій під будь-які потреби. Сценарій не лише впорядковує встановлення, але й дає змогу ефективно керувати встановленим програмним забезпеченням, роблячи процес максимально зручним і надійним, та ще формує папку з автономними пакетами програм (див. рис.3.26-3.28), які можна буде використати вже для інших сценаріїв.

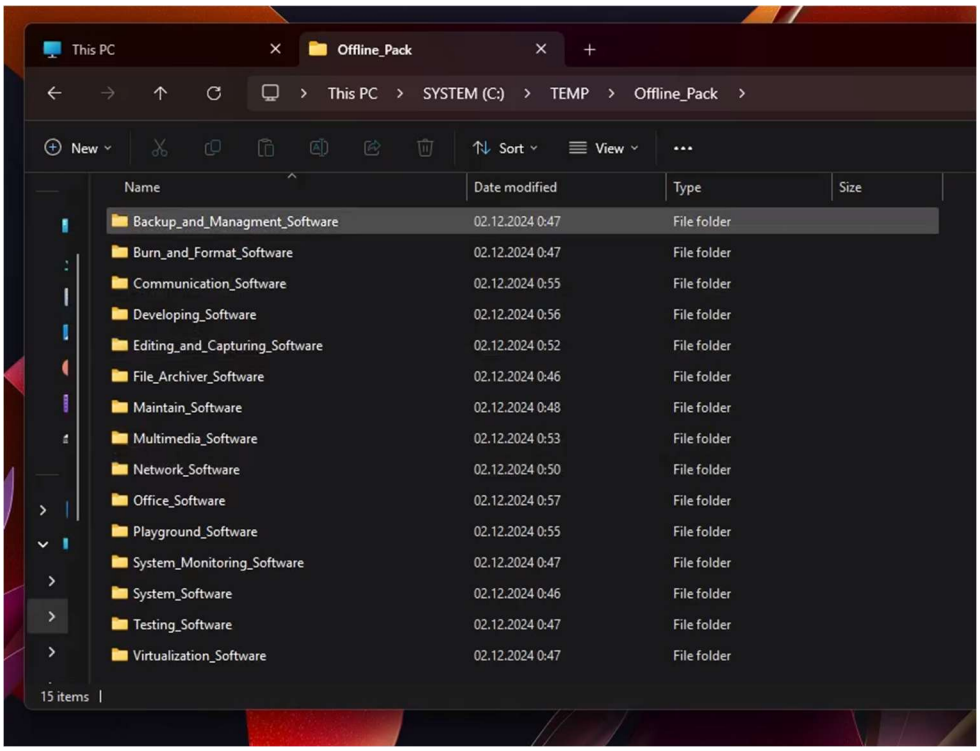

Рисунок 3.26 – Під час роботи сценарію створюється папка TEMP, куди завантажуються усі пакети програм

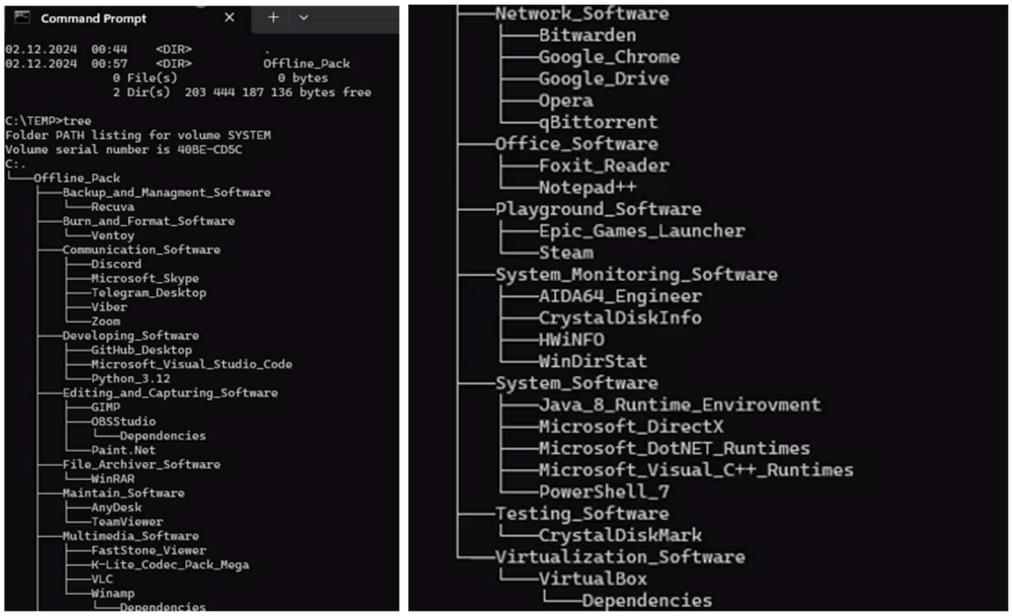

Рисунок 3.27 – Дерево папки, куди завантажуються усі пакети програм (для встановлення чи збереження на ПК, звідки був запущений скрипт)

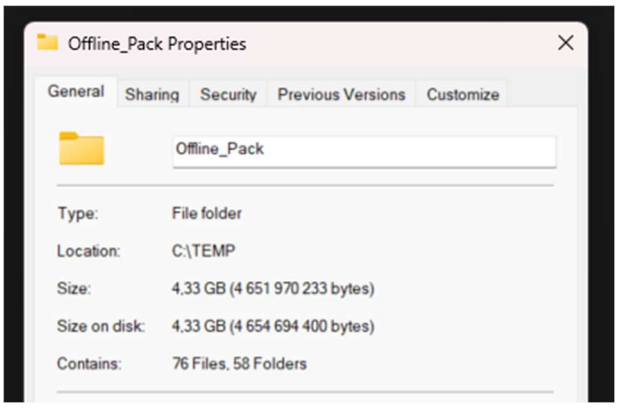

Рисунок 3.28 – Розмір автономного пакета програм,
що сформувався у папці TEMP

Якщо обрати іншу дію, тобто відмовитись від відображення (див. рис.3.29), сценарій виконається без показу пакета програм на екрані та одразу розпочнеться процес інсталяції програмного забезпечення, починаючи з першої підкатегорії «System Software» і закінчуючи п'ятнадцятою – «Office Software».

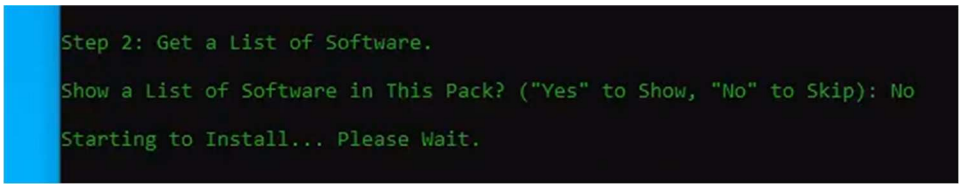

Рисунок 3.29 – У разі відмови про вивід пакету програм, відбувається
завантаження та/або встановлення пакету ПЗ

Після початку процесу, чи лише завантаження пакета програм або завантаження та встановлення, можемо побачити хід виконання (див. рис.3.30). У даному випадку відбувається встановлення пакета програм на нову, ще не налаштовану операційну систему, яка потребує першочергового налаштування. Використовуючи такий скрипт, можна спокійно займатися іншими справами, поки відбувається автоматичне встановлення необхідного для першочергового налаштування пакета програм.

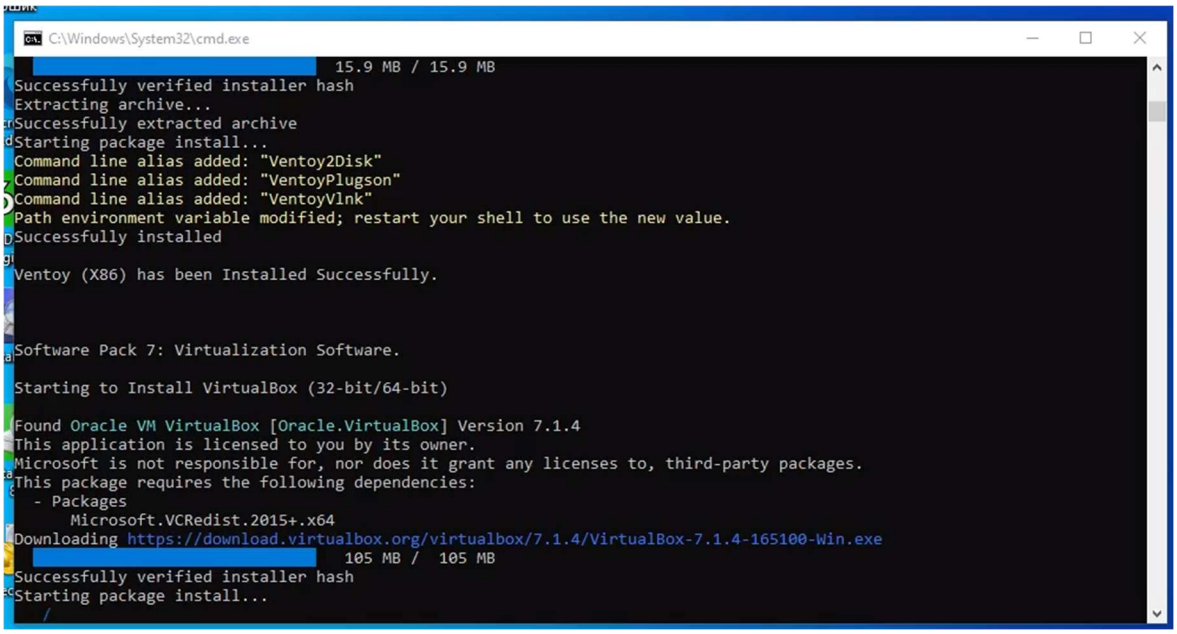

Рисунок 3.30 – Процес завантаження або/та встановлення пакету ПЗ

Це особливо зручно для налаштування, наприклад, ПК віддаленого працівника, який отримав новий пристрій. Завдяки такому сценарію, можемо налаштувати йому весь ПК, причому зможемо навіть користуватися ПК під час встановлення або займатися своїми справами. Адже запустивши цей сценарій – він виконується самостійно до повного встановлення ПЗ.

Основні підпрограми запропонованого сценарію (див. рис.3.31):

1. **:SAMPLE_PACK** – відображає структуру та вміст програмних пакетів.
2. :**OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE** – завантажує або встановлює ПЗ за допомогою Winget, перевіряє результат.
3. **:RENAME_PACKAGEFILE_WITH_CHECK_CYCLE** – виконує перейменування інсталяційних файлів.
4. **:ERROR_END** – виводить повідомлення про помилку та завершує роботу.
5. **:SUCCESS_END** – виводить повідомлення про успіх.

Кожен етап встановлення включає обробку помилок. Якщо інсталяція або завантаження певного компоненту завершуються невдало, це фіксується виведенням повідомлення про помилку. Успішно завантажені файли в офлайн-режимі зберігаються у відповідних директоріях, імена файлів оновлюються відповідно до патерну, що виконуються через виклики допоміжних підпрограм: ":OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE" для завантаження та :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE для перейменування файлів у потрібний формат (див. рис.3.31).

```
goto :eof

:OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE

winget %script_execution_action% !id_package! -d C:\TEMP\Offline_Pack\!destination_folder!\ & echo.  !!
if !errorlevel! equ 0 (
    echo !name_package! has been %script_execution_action%ed Successfully. & echo. & echo.
) else (
    echo !name_package! Failed to %script_execution_action%. & echo. & echo.
)
goto :eof


:RENAME_PACKAGEFILE_WITH_CHECK_CYCLE

cd /d "C:\TEMP\Offline_Pack\!destination_folder!" > nul 2>&1
del "C:\TEMP\Offline_Pack\!destination_folder!\*.yaml" /s /q > nul 2>&1
for /f "delims=" %%i in ('dir /b "*!arch_packagefile!*.!type_packagefile!"') do (
    ren "%%i" "!newname_packagefile!.!type_packagefile!"
    if !errorlevel! equ 0 (
        echo !secondname_packagefile! Successfully Renamed to !newname_packagefile!.!type_packagefile!.
    ) else (
        echo !secondname_packagefile! Not Found.
    )
)
goto :eof

```

Рисунок 3.31 – Звернення до підпрограм задля зменшення розмірності сценарію (головні підпрограми, що виконують завантаження або/та встановлення пакету програм, створюючи при цьому структуру за вказаним патерном)

Кожна така підпрограма містить певні зміни, які постійно коригуються відповідно до комбінації, вказаної в «тілі» сценарію (див. рис.3.32). Тобто, згідно з прописаними змінними, які потрібно змінити, ця змінна за допомогою "Call' підключається до підпрограми та вставляє змінні у підпрограму – цей процес триває до тих пір, поки програми не будуть лише завантажені або завантажені та встановлені на пристрій.

```
533
534  :: Obtain K-Lite Codec Pack Mega (32-bit and 64-bit Package Files)
535
536  echo Starting to %script_execution_action% K-Lite Codec Pack Mega ^(32-bit/64-bit^) & echo.
537  for %%a in (%cpu_arch_current%) do (
538      set "name_package=K-Lite Codec Pack Mega (X%%a)"
539      set "id_package=CodecGuide.K-LiteCodecPack.Mega -a x%%a"
540      set "destination_folder=Multimedia_Software\K-Lite_Codec_Pack_Mega"
541      call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
542      set "secondname_packagefile=K-Lite Codec Pack Mega (X%%a)"
543      set "newname_packagefile=K-Lite_Codec_Pack_Mega_X%%a"
544      set "arch_packagefile=%%a"
545      set "type_packagefile=exe"
546      call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
547  )
548
```

Рисунок 3.32 – Перед зверненням до підпрограми відбувається присвоєння змінним нових значень

```
890
891  :ERROR_END
892  echo.
893  echo Please Restart the Script or Try Again Later.
894  goto :REAL_END
895
896  :SUCCESS_END
897
898  echo.
899  echo Done.
900  goto :REAL_END
901
902  :REAL_END
903
904  echo.
905  pause
906  exit
```

Рисунок 3.33 – Варіанти завершення роботи сценарію (невірно введено відповідь; успішне завершення після виконання «тіла» скрипта)

Наприкінці скрипт інформує користувача про успішне завершення виконання (див. рис.3.34).

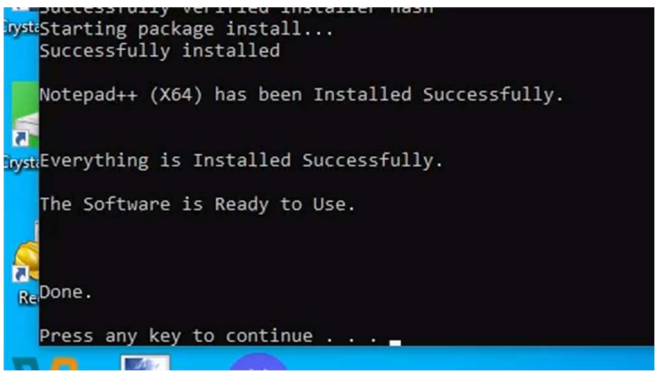

Рисунок 3.34 – Сценарій успішно виконав «тіло» сценарію

В результаті отримуємо повністю готовий та налаштований програмним забезпеченням ПК (див. рис.3.35), який може використовувати працівник або будь-яка інша людина, якій потрібно було встановити це програмне забезпечення.

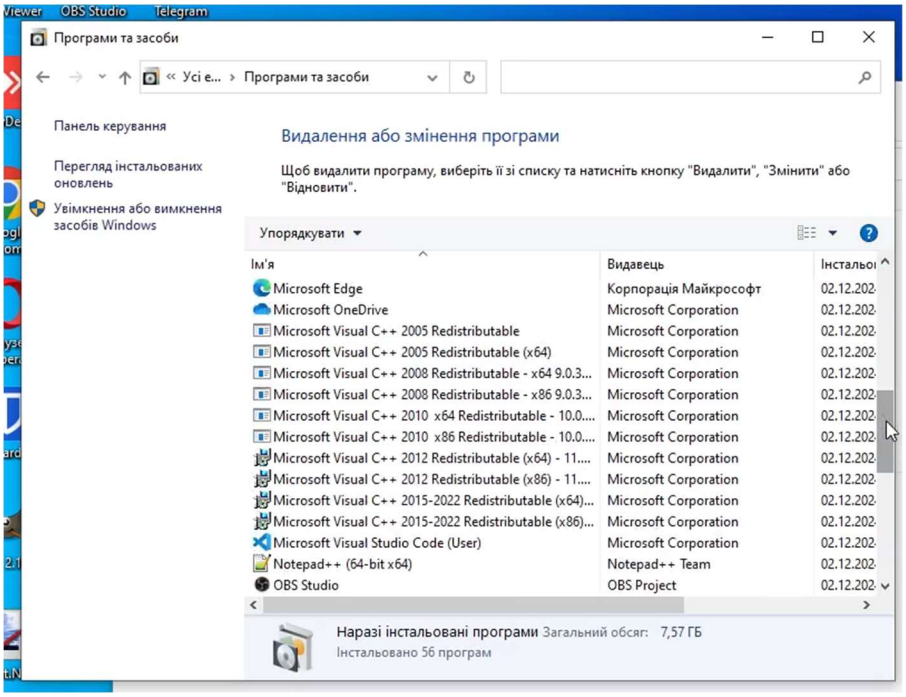

Рисунок 3.35 – Пакет ПЗ був успішно розвернутий на новій ОС (встановлено 42 програми)

Запропонований сценарій можна використовувати навіть при першому запуску ПК після покупки в магазині. Однак, якщо розглядати більш масштабно, такий скрипт можна запускати на великій кількості комп'ютерів, головне – наявність Інтернет-з'єднання.

Все виконується повністю автоматично та досить швидко, приблизно за 20-25 хвилин (див. рис.3.36-3.37), залежно від швидкості інтернет-з'єднання. Спочатку відбувається завантаження інсталяційних пакетів програм, після чого Winget забезпечує їх автоматичне встановлення, що значно спрощує та пришвидшує процес. Все встановлюється в автоматичному режимі, непомітно для користувача. Вручну встановлення програмного забезпечення з офіційних сайтів зайняло б не одну годину, щонайменше одну-дві, щоб встановити все необхідне. Натомість, даний сценарій виконує це все автоматично.

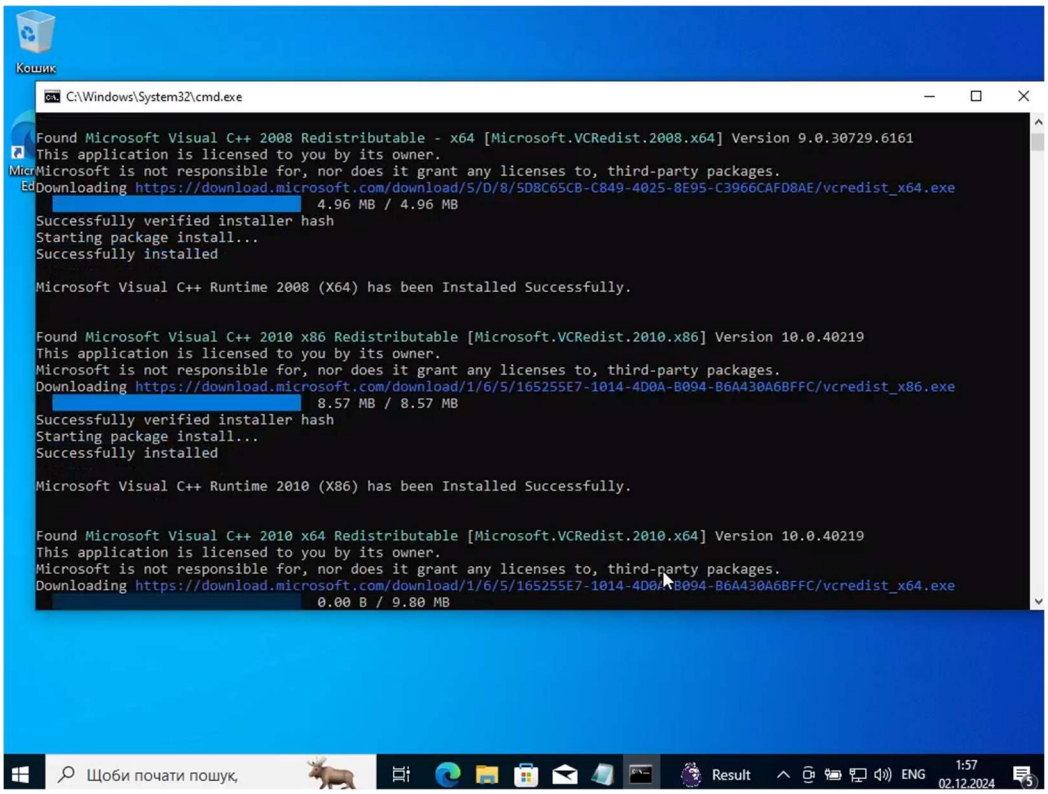

Рисунок 3.36 – Початок роботи сценарію безперервоного розгортання ПЗ (дата та час початку: 02.12.2024 – 1:57)

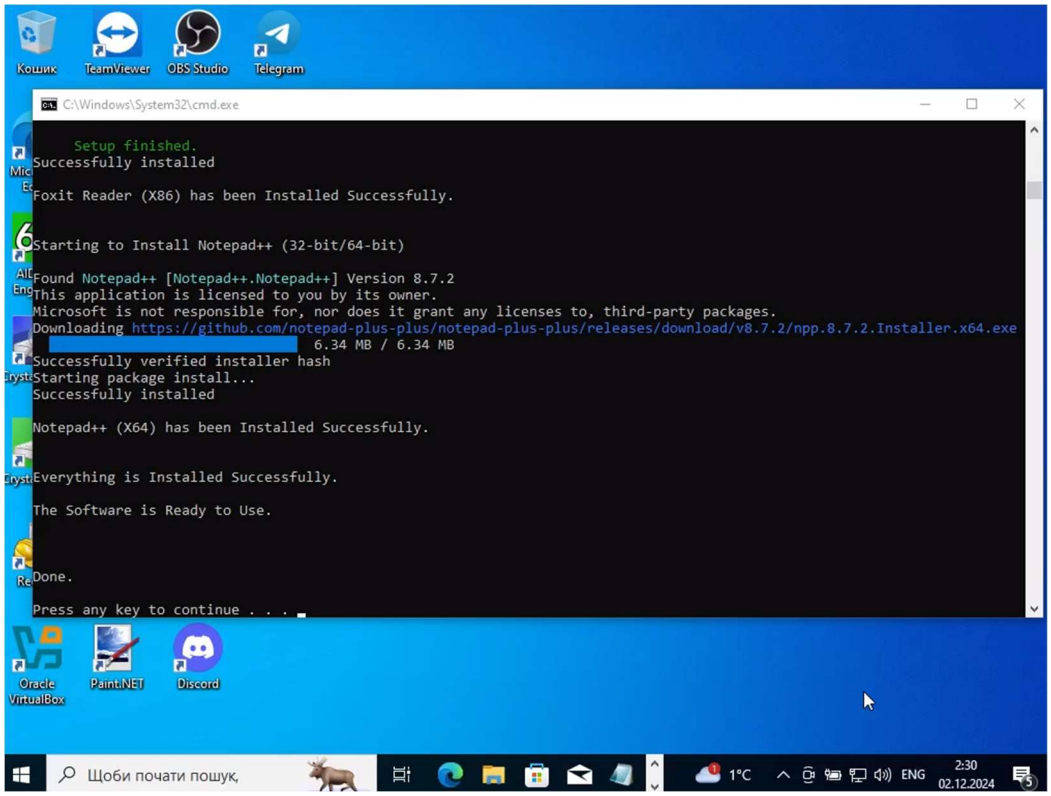

Рисунок 3.37 – Закінчення роботи сценарію безперервоного розгортання ПЗ (дата та час закінчення: 02.12.2024 – 2:30)

У підсумку, написана «міні-програма» виконала завантаження та встановлення ПЗ на нову ОС за 23 хв (див. рис.3.30-3.31), що надає цьому сценарію практичне застосування у потребі швидкого розгортання ПЗ, чим є потужним інструментом для централізованого обслуговування систем, забезпечуючи актуальність необхідного ПЗ з мінімальними зусиллями з боку користувача.

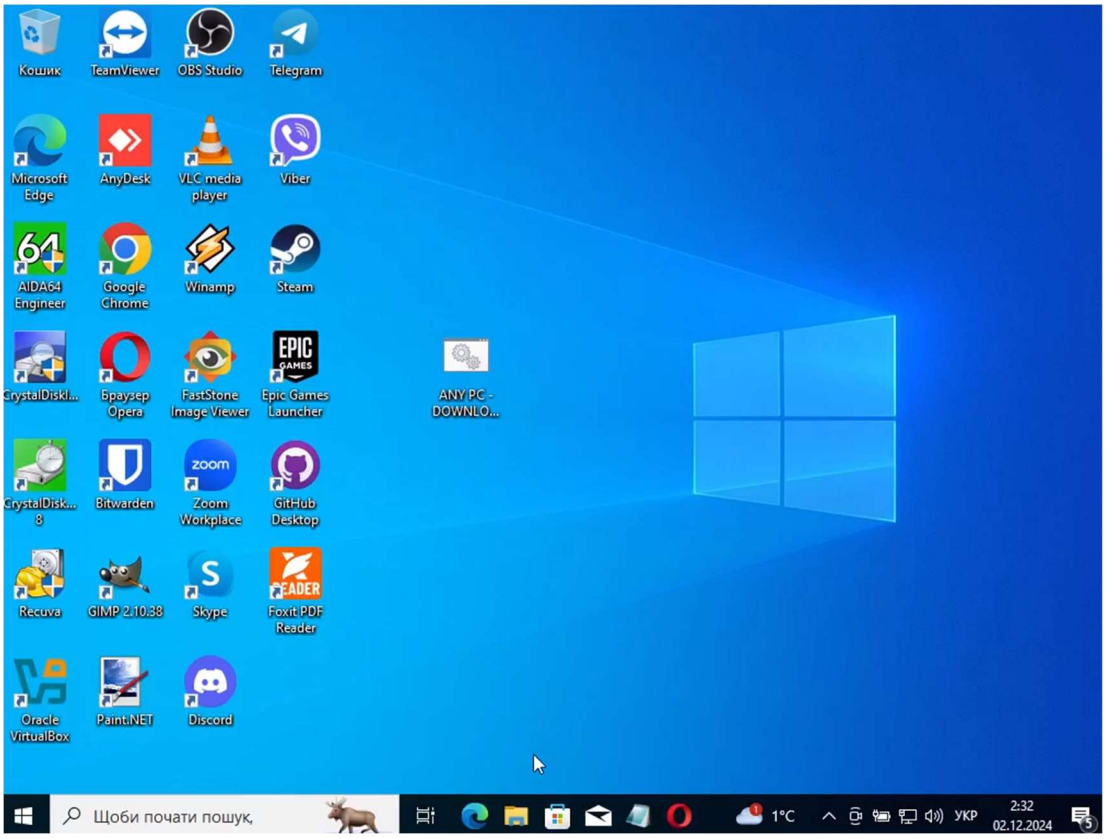

Рисунок 3.38 – Відображено робочий стіл зі розгорнутим пакетом ПЗ

3.3 Економічний ефект від впровадження автоматизованої системи

Аналізуючи сучасні можливості для управління кінцевими пристроями та автоматизації їх технічного обслуговування, такі як Microsoft Intune, Microsoft Entra чи інші платформи класу MDM, стає очевидним, що ці рішення пропонують значний спектр функцій. Однак, попри свою популярність, вони мають обмеження, які можуть впливати на ефективність впровадження, особливо для компаній, які прагнуть мінімізувати витрати та забезпечити максимальну кастомізацію під власні потреби. У цьому контексті сценарне рішення виступає як інноваційний підхід, який дозволяє досягти високого рівня автоматизації без необхідності значних інвестицій у ліцензії чи залежності від постачальників.

Однією з найбільш очевидних переваг сценарного підходу є його економічна ефективність. Комерційні рішення зазвичай вимагають регулярних витрат на ліцензування для кожного пристрою або користувача, а також на використання хмарної інфраструктури. Наприклад, для організації зі штатом у 1000 працівників загальні витрати на такі системи можуть досягати десятків тисяч доларів на рік. Натомість запропоноване сценарне рішення базується на використанні відкритих інструментів, таких як Windows Task Scheduler, PowerShell або Bash. Це дозволяє значно знизити витрати, обмежившись лише початковими вкладеннями на розробку та налаштування. Така модель не лише забезпечує швидку окупність, але й мінімізує залежність від сторонніх сервісів, знижуючи ризики неочікуваного збільшення витрат у майбутньому.

Гнучкість та адаптивність є ще одним визначальним фактором – комерційні платформи часто пропонують стандартизований набір функцій, що не завжди дозволяє врахувати специфічні потреби організації. У випадку сценарного підходу, організація має змогу створювати індивідуальні алгоритми та процеси, які ідеально відповідають її вимогам. Це стосується як особливостей IT-інфраструктури, так і специфіки роботи працівників, їхніх пристроїв та навіть умов використання. Наприклад, сценарії можуть бути налаштовані для виконання унікальних завдань,

таких як інтеграція нестандартного обладнання, забезпечення сумісності з кількома операційними системами або адаптація до змін у робочому середовищі.

Додатковою перевагою сценарного рішення є його автономність – на відміну від платформ, які залежать від постійного підключення до хмарних сервісів, сценарії можуть функціонувати локально, що значно підвищує безпеку даних, що особливо важливо для організацій, які працюють з чутливою інформацією, такою як фінансові установи чи компанії, що діють у сфері охорони здоров'я. Локальна робота сценаріїв дозволяє уникнути залежності від інтернет-з'єднання, що може бути критичним у віддалених регіонах чи під час тимчасових технічних проблем із мережею.

Ще одним аспектом, який вирізняє сценарний підхід, є його можливість оптимізувати використання програмного забезпечення. За допомогою інтегрованих аналітичних інструментів організація може отримати детальний огляд активності кожної програми, що встановлена на пристроях працівників, що дозволяє відмовлятися від непотрібних ліцензій, спрямовуючи ресурси на підтримку лише тих програм, які реально використовуються. Наприклад, якщо аналіз покаже, що певний софт рідко запускається або має дешевші альтернативи, організація може оптимізувати свої витрати, заощаджуючи значні кошти на ліцензуванні.

Масштабованість є ще однією вагомою перевагою – стандартні MDM-рішення зазвичай мають обмеження на кількість пристроїв чи користувачів, що відповідає обраному тарифному плану. У разі розширення організації ці обмеження можуть спричинити додаткові витрати або потребу в переході на дорожчі тарифи. Натомість сценарії не мають подібних обмежень. Вони дозволяють легко додавати нові пристрої чи користувачів без необхідності вносити зміни до структури системи або збільшувати витрати, що забезпечить простоту масштабування, що є особливо важливим для швидкозростаючих компаній.

Запобігання простоїв систем є ще одним важливим аспектом, де сценарії демонструють свою ефективність. Стандартні MDM-рішення зазвичай автоматизують процес оновлення та виправлення помилок, однак цей процес часто обмежується загальними налаштуваннями. У сценарному рішенні можливе

локальне тестування оновлень перед їх розгортанням, що мінімізує ризик конфліктів чи збоїв, що дозволить уникнути ситуацій, коли оновлення призводить до тимчасової непрацездатності пристрою, що, у свою чергу, знижує простої та втрати продуктивності.

Нарешті, незалежність від постачальників є вирішальним фактором для багатьох організацій. Багато комерційних платформ створюють залежність, яка обмежує можливості організації перейти на альтернативні рішення. Сценарний підхід повністю уникає цієї проблеми, оскільки базується на стандартних технологіях, які легко інтегруються з будь-якою інфраструктурою.

Таким запропоноване рішення забезпечує оптимальне поєднання економічності, гнучкості, автономності, масштабованості та незалежності. Воно дозволяє організаціям використовувати сучасні підходи до автоматизації без необхідності великих витрат, зберігаючи високий рівень продуктивності, безпеки та адаптивності до змін. У порівнянні зі стандартними платформами, сценарне рішення є більш універсальним і вигідним вибором для компаній, що прагнуть ефективно управляти своїми IT-ресурсами.

Таблиця 3.1 – Порівняння запропонованого рішення між комерційними

Критерій	**Запропоноване рішення (автоматизований сценарій)**	**Комерційні-рішення (Microsoft Intune, Entra тощо)**
Вартість	Початкові інвестиції, відсутність ліцензійних платежів.	Постійні витрати на ліцензії, що залежать від кількості користувачів чи пристроїв.
Гнучкість налаштувань	Максимальна адаптивність, налаштування під конкретні потреби організації.	Обмежена функціональність, стандартизовані підходи.

Продовження таблиця 3.1 – Порівняння запропонованого рішення між комерційними

Критерій	**Запропоноване рішення (автоматизований сценарій)**	**Комерційні-рішення (Microsoft Intune, Entra тощо)**
Автономність	Працює локально, не залежить від постійного інтернет-з'єднання чи хмарних сервісів. Інтернет використовується лише для завантаження необхідних пакетів ПЗ.	Залежність від хмарних серверів та інтернет-з'єднання.
Швидкість розгортання	Встановлення 42 програм за 23 хвилини. Ручний процес зайняв би понад годину.	Час залежить від хмарних серверів та доступності інтернету.
Масштабованість	Легке масштабування без додаткових витрат.	Додаткові витрати на масштабування та зміну тарифних планів.
Захист даних	Локальна обробка, що зменшує ризики витоку інформації.	Дані зберігаються в хмарі, ризик витоків чи залежність від політик постачальника.
Залежність від постачальника	Повна незалежність, використання відкритих технологій.	Висока залежність від конкретного постачальника і його екосистеми.
Оптимізація витрат на ПЗ	Автоматична аналітика для управління ліцензіями, відмова від зайвих програм.	Можливості оптимізації обмежені стандартними звітами.
Контроль оновлень	Локальне тестування оновлень, мінімізація ризиків простоїв.	Автоматичне розгортання, але ризики конфліктів залишаються.

ВИСНОВКИ

Автоматизація технічного обслуговування кінцевих пристроїв у сучасному бізнес-середовищі відкриває нові можливості для ефективного управління ресурсами та оптимізації процесів. Пропоноване рішення забезпечує швидке розгортання ПЗ, демонструючи значні переваги у швидкості виконання завдань. Зокрема, можливість встановлювати 42 програми за 23 хвилини, що значно перевершує понад годину, необхідну для ручного процесу, робить це рішення максимально практичним і економічно вигідним.

Ключовою перевагою є автономність, що дозволяє системі працювати без постійного підключення до інтернету. Завантаження необхідних пакетів ПЗ відбувається лише один раз, що мінімізує залежність від зовнішніх факторів, таких як мережеві збої чи нестабільне інтернет-з’єднання, що забезпечує високу надійність навіть у віддалених або малодоступних місцях, де доступ до мережі є обмеженим.

Продуктивність і безпека системи підвищуються завдяки централізованому керуванню пристроями, швидкому оновленню програмного забезпечення та зниженню ручної праці. Такий підхід мінімізує ризик людських помилок, скорочує час простоїв і захищає інфраструктуру від кіберзагроз завдяки оперативному впровадженню патчів безпеки. Масштабованість рішення дозволяє ефективно інтегрувати нові пристрої та користувачів, зменшуючи витрати на додатковий персонал.

Потенціал автоматизації може зрости у майбутньому зі впровадженням сучасних технологій, таких як штучний інтелект і машинне навчання, які б дозволили прогнозувати можливі збої, запобігати їм завчасно та автоматизувати управління всією інфраструктурою. Таким чином, система стає ключовим елементом у цифровій трансформації бізнесу, відповідаючи вимогам як сьогодення, так і майбутнього розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Remote Worker Wave Is Here – Is Your Network Ready?. Network Coverage - Managed IT Service Provider. URL: https://www.netcov.com/on-premise-network-changes-needed-for-remote-work/ (дата звернення: 15.10.2024).
2. Should Your Employees Use Their Personal Devices for Work? Best Practices for BYOD. Phone.com. URL: https://www.phone.com/should-your-employees-use-their-personal-devices-for-work-best-practices-for-byod/ (дата звернення: 15.10.2024).
3. BYOD vs. corporate-owned smartphones: What's better for small business?. Samsung Business Insights. URL: https://insights.samsung.com/2022/06/28/byod-vs-corporate-issued-smartphones-which-is-better-for-small-business-3/ (дата звернення: 15.10.2024).
4. Sencio P. What Are Remote Devices? Their Importance for IT Management. InvGate ITSM blog - ITAM, ESM, ITIL, and more. URL: https://blog.invgate.com/remote-devices (дата звернення: 15.10.2024).
5. Yasar K., Gillis A. S. What is IoT (Internet of Things)? | Definition from TechTarget. Search IoT. URL: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT (дата звернення: 15.10.2024).
6. QA for the Internet of Things: A guide to implementing an IoT testing framework. Yalantis. URL: https://yalantis.com/blog/iot-testing-guide/ (дата звернення: 18.10.2024).
7. The Future of Work: How Virtual Desktops are Transforming Remote Work. Apporto. URL: https://www.apporto.com/the-future-of-work-how-virtual-desktops-are-transforming-remote-work (дата звернення: 18.10.2024).
8. What is enterprise networking?. Cloudflare. URL: https://www.cloudflare.com/learning/network-layer/enterprise-networking/ (дата звернення: 18.10.2024).
9. Why Are Software Updates and Patches Important? - Agio. Agio - Technology support and data protection. When and where you need it. URL: https://agio.com/why-is-it-important-to-keep-software-updated/#gref (дата звернення: 18.10.2024).
10. How do you automate system updates?. LinkedIn: Log In or Sign Up. URL: https://www.linkedin.com/advice/0/how-do-you-automate-system-updates-skills-system-administration (дата звернення: 18.10.2024).
11. Simple Guide to Failure Metrics (MTBF vs. MTTR vs. MTTF). Resco. URL: https://www.resco.net/learning/failure-metrics/#:~:text=The%20total%20downtime%20caused%20by,MTTR%20will%20be%20two%20hours. (дата звернення: 20.10.2024).
12. Future of Automation in Maintenance. Sensemore. URL: https://sensemore.io/future-of-automation-in-maintenance/ (дата звернення: 20.10.2024)

13. Remote maintenance: Which tools should you choose? - SEP - Systancia Experience Portal. SEP - Systancia Experience Portal. URL: https://www.systancia.com/en/remote-maintenance-which-tools-should-you-choose/ (дата звернення: 20.10.2024).
14. Microsoft Intune. glueckkanja. URL: https://www.glueckkanja.com/en/modern-workplace/microsoft-intune/ (дата звернення: 21.10.2024).
15. Zabbix vs. Nagios: Which one to choose?. Hawatel. URL: https://hawatel.com/en/blog/zabbix-vs-nagios-which-one-to-choose/ (дата звернення: 21.10.2024).
16. What is Splunk? Key Benefits and Features of Splunk. Fortinet. URL: https://www.fortinet.com/resources/cyberglossary/what-is-splunk (дата звернення: 23.10.2024).
17. What is mobile device management (MDM) - CNC. CNC. URL: https://cnc-ltd.co.uk/2023/01/03/what-is-mobile-device-management-mdm/ (дата звернення: 23.10.2024).
18. Godfrey P. Empowering Secure and Efficient Device Management: Why Use Microsoft Intune?. LinkedIn: Log In or Sign Up. URL: https://www.linkedin.com/pulse/empowering-secure-efficient-device-management-why-use-paul-godfrey/ (дата звернення: 23.10.2024).
19. Endpoint Verification overview | Endpoint Verification Documentation | Google Cloud. Google Cloud. URL: https://cloud.google.com/endpoint-verification/docs/overview (дата звернення: 25.10.2024).
20. What is Microsoft Entra ID?. Training | Microsoft Learn. URL: https://learn.microsoft.com/en-us/entra/fundamentals/whatis (дата звернення: 25.10.2024).
21. Comparison of AWS, Azure and Google Cloud for Backup. MSP360 Blog. URL: https://www.msp360.com/resources/blog/comparison-aws-azure-google/ (дата звернення: 25.10.2024).
22. Replace Cisco Umbrella | DNS Security. Connect, Protect and Build Everywhere | Cloudflare. URL: https://www.cloudflare.com/zero-trust/compare/cloudflare-vs-cisco-umbrella/ (дата звернення: 25.10.2024).
23. Unified Endpoint Management Tools. Garther. URL: https://www.gartner.com/reviews/market/unified-endpoint-management-tools (дата звернення: 25.10.2024).
24. Systemsempolyee. Intelligent Sensor Data Platforms in the Internet of Things (IoT) Era. Medium. URL: https://medium.com/@systemsempolyee/intelligent-sensor-data-platforms-in-the-internet-of-things-iot-era-f459aab1ea01 (дата звернення: 26.10.2024).
25. Remote Monitoring Using IoT for Real-Time Insights. PsiBorg Technologies Pvt. Ltd. URL: https://psiborg.in/applications-of-industrial-remote-monitoring-using-iot/ (дата звернення: 27.10.2024).
26. PcSite. Integrating IoT Devices with Your PC Server. Medium. URL: https://pcsite.medium.com/integrating-iot-devices-with-your-pc-server-826f6a8d8ece (дата звернення: 27.10.2024).

27. AWS IoT vs. Azure IoT: Choose the Best IoT Platform. Rishabh Software. URL: https://www.rishabhsoft.com/blog/aws-iot-vs-azure-iot-comparison (дата звернення: 27.10.2024).
28. Difference Between CMD vs Powershell vs Bash - AttuneOps. AttuneOps. URL: https://attuneops.io/difference-between-cmd-vs-powershell-vs-bash/#:~:text=CMD%20is%20the%20command%20line,/Linux-based%20operating%20systems. (дата звернення: 27.10.2024).
29. Sharma R. Python for Automation: Streamlining Tasks with Scripting. Medium. URL: https://medium.com/pythoneers/python-for-automation-streamlining-tasks-with-scripting-1230be4790cf (дата звернення: 27.10.2024).
30. Use the WinGet tool to install and manage applications. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows/package-manager/winget/ (дата звернення: 28.10.2024).
31. How Chocolatey Works. Chocolatey Software. URL: https://chocolatey.org/how-chocolatey-works (дата звернення: 28.10.2024).
32. Remote Desktop Manager. Devolutions. URL: https://devolutions.net/remote-desktop-manager/ (дата звернення: 28.10.2024).
33. Intune and Configuration Manager (MECM) (SCCM) | Finding the Right Tools for Unified Endpoint Management. Mobile Mentor. URL: https://www.mobile-mentor.com/insights/microsoft-endpoint-manager-intune-and-configuration-manager-sccm/ (дата звернення: 28.10.2024).
34. (MDM) Installing the agent using Microsoft Intune. URL: https://help.forcepoint.com/endpoint/en-us/onlinehelp/guid-56bc6033-169e-4c7b-a971-0f9ab049a2ab.html (дата звернення: 28.10.2024).
35. Microsoft Azure - MongoDB Atlas. MongoDB: The Developer Data Platform | MongoDB. URL: https://www.mongodb.com/docs/atlas/reference/microsoft-azure/ (дата звернення: 29.10.2024).
36. GitHub - tinyplayerss/Winget-Wizard: Winget Wizard Tool. GitHub. URL: https://github.com/tinyplayerss/Winget-Wizard (дата звернення: 29.10.2024).
37. GitHub - ChrisTitusTech/winutil: Chris Titus Tech's Windows Utility - Install Programs, Tweaks, Fixes, and Updates. GitHub. URL: https://github.com/ChrisTitusTech/winutil (дата звернення: 29.10.2024).
38. Google Workspace vs Office 365: The Ultimate Business Solution Comparison in 2024. Medha Cloud: Multi Cloud & Managed IT Service Provider. URL: https://medhacloud.com/blog/google-workspace-vs-office-365-business-2024/#:~:text=Gmail%20work%20perfectly.-,Security%20and%20compliance:%20Keeping%20your%20business%20safe,protection%20against%20phishing%20and%20spam. (дата звернення: 30.10.2024).
39. Products. Yubico. URL: https://www.yubico.com/products/ (дата звернення: 30.10.2024).
40. About Windows Remote Management - Win32 apps. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows/win32/winrm/about-windows-remote-management (дата звернення: 30.10.2024).

41. What Is A SIEM? Benefits, Tools, & Strategies. PurpleSec. URL: https://purplesec.us/learn/siem-solutions/ (дата звернення: 30.10.2024).

42. Revista de Investigación Científica Huamachuco. Impact of software testing automation on the development cycle. ResearchGate. URL: https://www.researchgate.net/publication/375100440_Impact_of_software_testing_automation_on_the_development_cycle (дата звернення: 30.10.2024).

43. Гафанович В.О., Антоненко А.В. Впровадження системи електронного документообігу та управління даними в корпоративній мережі.«Застосування програмного забезпечення в ІКТ» : Всеукр. науково-техн. конф., м.Київ, 24 квітня 2024 р. С. 180-182. URL: https://duikt.edu.ua/uploads/p_2661_45497999.pdf.

44. BUILDING CORPORATE NETWORKS FOR FOOD INDUSTRY ENTERPRISES / A. BALVAK, M. DEMCHENKO, V. DRAHUN, V. PETRENKO, V. PRYSIAZHNIUK, O. IVANOV, S. KOROTKOV, O. TSVYK, O. TONKYKH, A. ANTONENKO, A. LEMESHKO, V. HAFANOVYCH, O. HOLUBENKO. European Science, (sge22-01), (2023), P.25–47. URL: https://doi.org/10.30890/2709-2313.2023-22-01-031

45. Гафанович В.О. Автоматизація управління програмним забезпеченням на Windows за допомогою Winget. «Проблеми комп’ютерної інженерії» : Науково-практ. конф., м.Київ, 3 грудня 2024 р. С.181-183.

46. Гафанович В.О. Використання сценарія для встановлення програмного забезпечення онлайн та офлайн режимах. «Проблеми комп’ютерної інженерії» : Науково-практ. конф., м.Київ, 3 грудня 2024 р. С.184-185.

Додаток А

Кошторис використання існуючих рішень автоматизації обслуговування пристроїв в умовах віддаленої роботи

Таблиця А.1 – Кошторис та основні функції існуючих рішень автоматизації обслуговування пристроїв в умовах віддаленої роботи

Рішення	Вартість за 1 пристрій/ місяць	Вартість за 100 пристроїв/ місяць	Вартість за 100 пристроїв/рік	Основні функції
Microsoft Intune	$6-$10	$600-$1000	$7200-$12,000	Управління Windows, iOS, Android. Інтеграція з Microsoft 365. Налаштування політик безпеки. Віддалене стирання даних. Захист корпоративних даних. Призначення додатків і політик користувачам.
Google MDM	$3-$6	$300-$600	$3600-$7200	Інтеграція з Google Workspace. Управління ChromeOS, Android. Автоматизація розгортання додатків. Захист пристроїв від загроз. Віддалене блокування.
Jamf Pro (для macOS/ iOS)	$10-$15	$1000-$1500	$12,000-$18,000	Управління пристроями Apple. Масове розгортання додатків. Налаштування політик безпеки. Інтеграція з Apple Business Manager. Можливість кастомізації пристроїв.
VMware Workspace ONE	$3.50-$9	$350-$900	$4200-$10,800	Підтримка Windows, macOS, Android, iOS. Інтеграція з vSphere. Уніфіковане управління додатками. Захист конфіденційності даних. Контроль стану пристроїв у реальному часі.
Cisco Meraki Systems Manager	$2-$5	$200-$500	$2400-$6000	Моніторинг пристроїв у хмарі. Налаштування мережевих параметрів. Управління політиками доступу. Інтеграція з мережами Meraki. Віддалена діагностика.
ManageEngine MDM	$2-$4	$200-$400	$2400-$4800	Управління Windows, macOS, iOS, Android. Налаштування політик безпеки. Віддалене встановлення додатків. Генерація детальних звітів. Автоматизація рутинних завдань.

Додаток Б

Демонстрація розробленої міні-програми (сценарію) для реалізації безперервного першочергового обслуговування пристроїв (лише завантаження / завантаження встановлення)

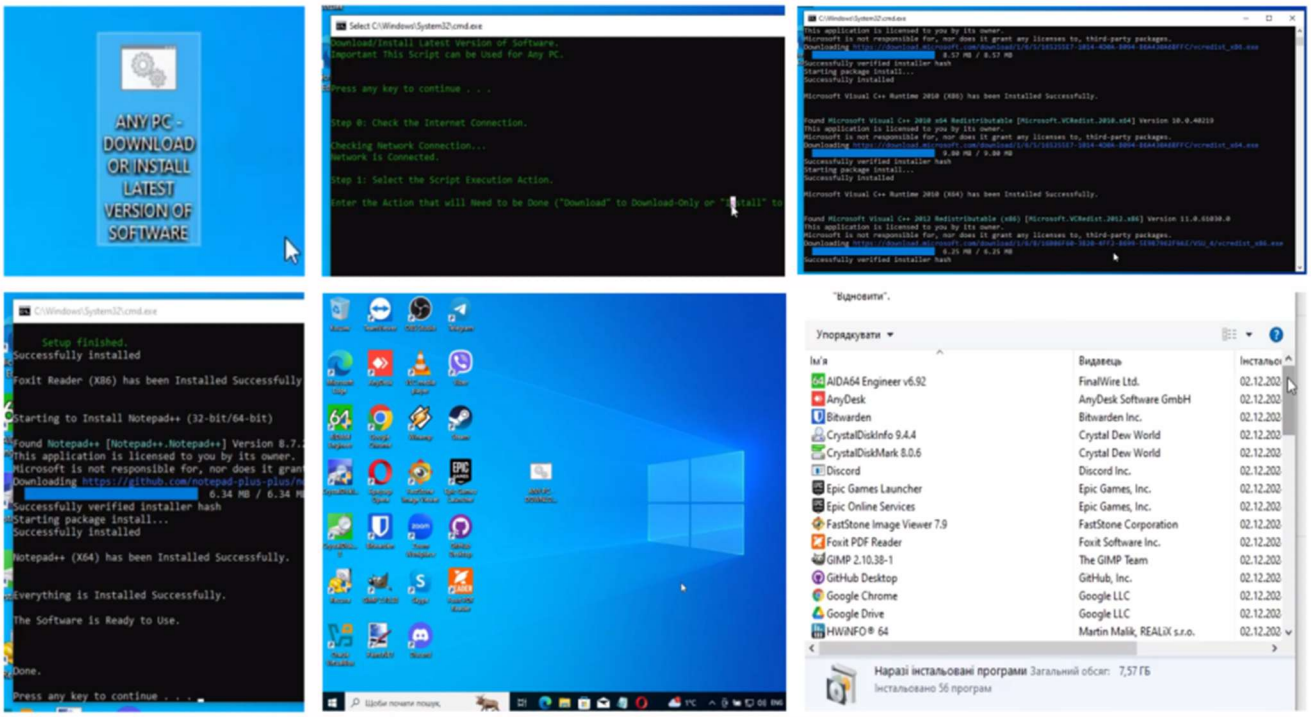

Рисунок Б.1 – Розроблена міні-програма (сценарій), що реалізує безперервне першочергове обслуговування пристроїв (завантаження та встановлення)

Продовження Додатка Б

Демонстрація розробленої міні-програми (сценарію) для реалізації безперервного першочергового обслуговування пристроїв (лише завантаження / завантаження встановлення)

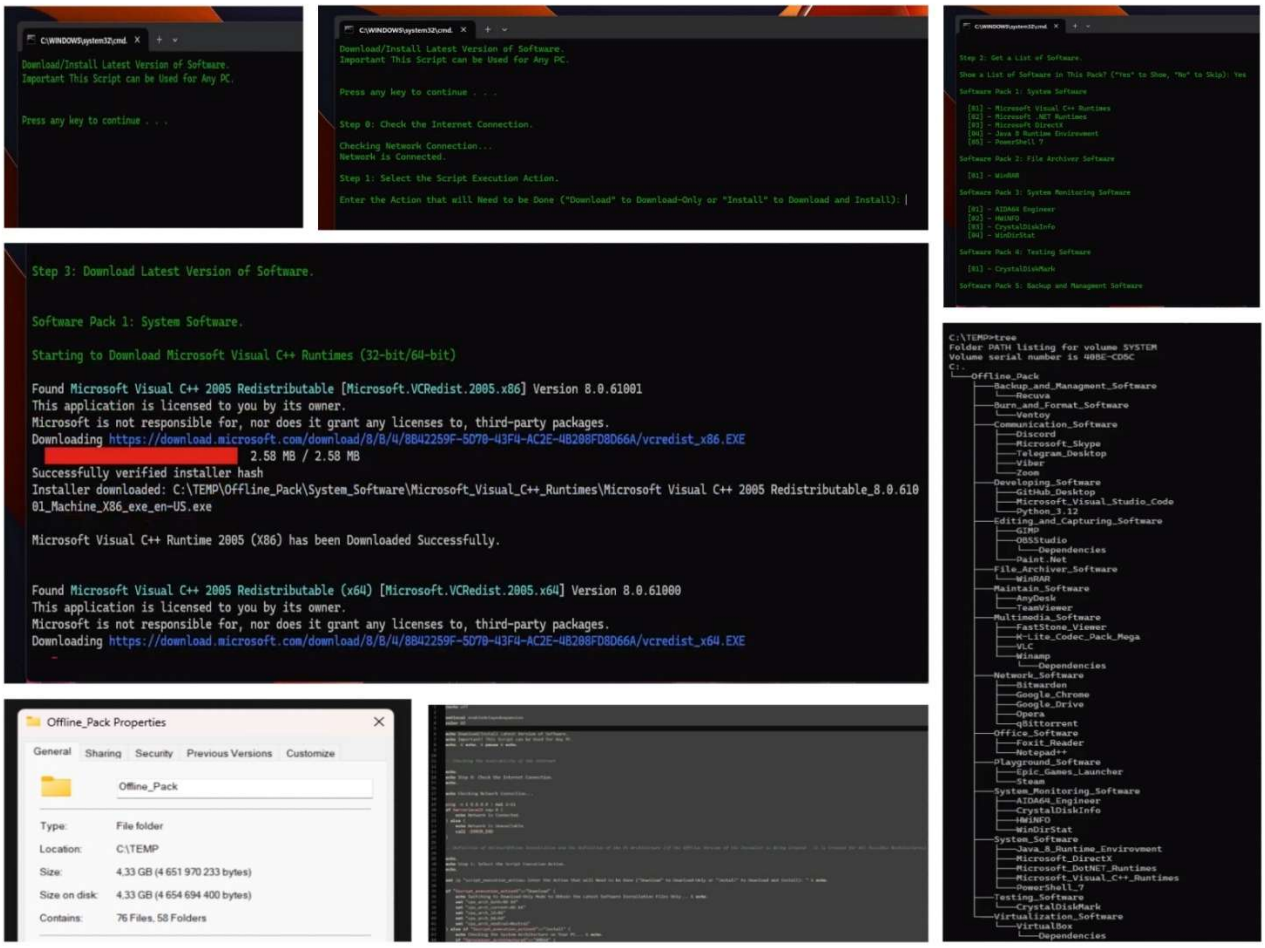

Рисунок Б.2 – Розроблена міні-програма (сценарій), що реалізує безперервне першочергове обслуговування пристроїв (лише завантаження)

Додаток В

Код розробленої міні-програми (сценарію) для реалізації безперервного першочергового обслуговування пристроїв

```
@echo off

setlocal enabledelayedexpansion
color 02

echo Download/Install Latest Version of Software.
echo Important! This Script can be Used for Any PC.
echo. & echo. & pause & echo.


:: Checking the Availability of the Internet

echo.
echo Step 0: Check the Internet Connection.
echo.

echo Checking Network Connection...

ping -n 1 8.8.8.8 > nul 2>&1
if %errorlevel% equ 0 (
    echo Network is Connected.
) else (
    echo Network is Unavailable.
    call :ERROR_END
)

:: Definition of Online/Offline Installation and the Definition of the PC
Architecture (if the Offline Version of the Installer is Being Created - it is
Created for All Possible Architectures)

echo.
echo Step 1: Select the Script Execution Action.
echo.

set /p "script_execution_action= Enter the Action that will Need to be Done
("Download" to Download-Only or "Install" to Download and Install): " & echo.

if "%script_execution_action%"=="Download" (
    echo Switching to Download-Only Mode to Obtain the Latest Software
Installation Files Only... & echo.
    set "cpu_arch_both=86 64"
    set "cpu_arch_current=86 64"
    set "cpu_arch_32=86"
    set "cpu_arch_64=64"
    set "cpu_arch_neutral=Neutral"
) else if "%script_execution_action%"=="Install" (
    echo Checking the System Architecture on Your PC... & echo.
    if "%processor_architecture%"=="AMD64" (
        set "cpu_arch_both=86 64"
        set "cpu_arch_current=64"
        set "cpu_arch_32=86"
        set "cpu_arch_64=64"
        echo System Architecture on %computername%: 64-bit  & echo. & echo.
    ) else (
        set "cpu_arch_both=86"
        set "cpu_arch_current=86"
        set "cpu_arch_32=86"
```

Продовження Додатка В

```
        set "cpu_arch_64=86"
        echo System Architecture on %computername%: 32-bit  & echo. & echo.
    )
    echo Switching to Install-Mode to Obtain and Install the Latest Software... &
echo.
) else (
    echo Incorrectly Entered Data. & echo.
    call :ERROR_END
)

:: Dropdown List of Software Included in this Script for Online/Offline
Installation

echo.
echo Step 2: Get a List of Software.
echo.

set /p "list_of_pack= Show a List of Software in This Pack? ("Yes" to Show, "No"
to Skip): " & echo.

if "!list_of_pack!"=="Yes" (
    call :SAMPLE_PACK
    set /p  "confirm_pack= Continue %script_execution_action% the Pack? ("Yes" to
Continue, "No" to Exit): " & echo.
    if "!confirm_pack!"=="Yes" (
        echo Starting to !script_execution_action!... Please Wait. & echo. & echo.
        call :MAIN
    ) else if "!confirm_pack!"=="No" (
        echo Exit From Script... Please Wait. & echo. & echo.
        call :SUCCESS_END
    ) else (
        echo Incorrectly Entered Data. & echo.
        call :ERROR_END
    )
) else if "!list_of_pack!"=="No" (
    echo Starting to !script_execution_action!... Please Wait. & echo. & echo.
    call :MAIN
) else (
    echo Incorrectly Entered Data. & echo.
    call :ERROR_END
)

:: The Process of Creating an Installer for Online/Offline Installation

:MAIN

echo.
echo Step 3: %script_execution_action% Latest Version of Software.
echo.

echo.
echo Software Pack 1: System Software.
echo.

:: Obtain Microsoft Visual C++ Runtimes (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Microsoft Visual C++ Runtimes ^(32-
bit/64-bit^) & echo.
for %%a in (2005 2008 2010 2012 2015+) do (
    for %%b in (%cpu_arch_both%) do (
        set "name_package=Microsoft Visual C++ Runtime %%a (X%%b)"
```

```
        set "id_package=Microsoft.VCRedist.%%a.x%%b"
        set "destination_folder=System_Software\Microsoft_Visual_C++_Runtimes"
        call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
        set "secondname_packagefile=Microsoft Visual C++ %%a %%b"
        set "newname_packagefile=Microsoft_Visual_C_Runtime_%%a_X%%b"
        set "arch_packagefile=%%b"
        set "type_packagefile=exe"
        call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
    )
)

:: Obtain Microsoft .NET Runtimes (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Microsoft .NET Runtimes ^(32-bit/64-
bit^) & echo.
for %%a in (3_1 5 6 7 8) do (
    for %%b in (%cpu_arch_both%) do (
        set "name_package=Microsoft .NET Runtime %%a (X%%b)"
        set "id_package=Microsoft.DotNet.Runtime.%%a -a x%%b"
        set "destination_folder=System_Software\Microsoft_DotNET_Runtimes"
        call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
        set "secondname_packagefile=Microsoft .NET Runtime %%a %%b"
        set "newname_packagefile=Microsoft_NET_Runtime_%%a_X%%b"
        set "arch_packagefile=%%b"
        set "type_packagefile=exe"
        call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
    )
)

:: Obtain Microsoft DirectX (Neutral Package File)

echo Starting to %script_execution_action% Microsoft DirectX ^(Neutral^) & echo.
for %%a in (%cpu_arch_neutral%) do (
    set "name_package=Microsoft DirectX (%%a)"
    set "id_package=Microsoft.DirectX"
    set "destination_folder=System_Software\Microsoft_DirectX"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Microsoft DirectX (%%a)"
    set "newname_packagefile=Microsoft_DirectX_%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Java 8 Runtime Envirovment (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Java 8 Runtime Envirovment ^(32-bit/64-
bit^) & echo.
for %%a in (%cpu_arch_both%) do (
    set "name_package=Java 8 Runtime Envirovment (X%%a)"
    set "id_package=Oracle.JavaRuntimeEnvironment -a x%%a"
    set "destination_folder=System_Software\Java_8_Runtime_Envirovment"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Java 8 Runtime Envirovment X%%a"
    set "newname_packagefile=Java_8_Runtime_Envirovment_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain PowerShell 7 (32-bit and 64-bit Package Files)
```

```
echo Starting to %script_execution_action% PowerShell 7 ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=PowerShell 7 (X%%a)"
    set "id_package=Microsoft.PowerShell -a x%%a"
    set "destination_folder=System_Software\PowerShell_7"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=PowerShell 7 X%%a"
    set "newname_packagefile=PowerShell_7_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=msi"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 2: File Archiver Software.
echo.

:: Obtain WinRAR (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% WinRAR ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=WinRAR (X%%a)"
    set "id_package=RARLab.WinRAR -a x%%a"
    set "destination_folder=File_Archiver_Software\WinRAR"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=WinRAR X%%a"
    set "newname_packagefile=WinRAR_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 3: System Monitoring Software.
echo.

:: Obtain AIDA64 Engineer (Only 32-bit Package File)

echo Starting to %script_execution_action% AIDA64 Engineer ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=AIDA64 Engineer (X%%a)"
    set "id_package=FinalWire.AIDA64.Engineer -a x%%a"
    set "destination_folder=System_Monitoring_Software\AIDA64_Engineer"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=AIDA64 Engineer X%%a"
    set "newname_packagefile=AIDA64_Engineer_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain HWiNFO (Only 64-bit Package File)

echo Starting to %script_execution_action% HWiNFO ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=HWiNFO (X%%a)"
    set "id_package=REALiX.HWiNFO -a x%%a"
    set "destination_folder=System_Monitoring_Software\HWiNFO"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
```

```
    set "secondname_packagefile=HWiNFO X%%a"
    set "newname_packagefile=HWiNFO_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain CrystalDiskInfo Only With "--accept-package-agreements" to Accept All
License Agreements For Packages Silenty (Only 32-bit Package File)

echo Starting to %script_execution_action% CrystalDiskInfo ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=CrystalDiskInfo (X%%a)"
    set "id_package=XP8K4RGX25G3GM -a x%%a --accept-package-agreements"
    set "destination_folder=System_Monitoring_Software\CrystalDiskInfo"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=CrystalDiskInfo X%%a"
    set "newname_packagefile=CrystalDiskInfo_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain WinDirStat (Only 32-bit Package File)

echo Starting to %script_execution_action% WinDirStat ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=WinDirStat (X%%a)"
    set "id_package=WinDirStat.WinDirStat -a x%%a"
    set "destination_folder=System_Monitoring_Software\WinDirStat"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=WinDirStat X%%a"
    set "newname_packagefile=WinDirStat_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 4: Testing Software.
echo.

:: Obtain CrystalDiskMark (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% CrystalDiskMark ^(32-bit/64-bit^) &
echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=CrystalDiskMark (X%%a)"
    set "id_package=CrystalDewWorld.CrystalDiskMark -a x%%a"
    set "destination_folder=Testing_Software\CrystalDiskMark"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=CrystalDiskMark X%%a"
    set "newname_packagefile=CrystalDiskMark_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)
```

```
echo.
echo Software Pack 5: Backup and Managment Software.
echo.

:: Obtain Recuva (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Recuva ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=Recuva (X%%a)"
    set "id_package=Piriform.Recuva -a x%%a"
    set "destination_folder=Backup_and_Managment_Software\Recuva"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Recuva X%%a"
    set "newname_packagefile=Recuva_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 6: Burn and Format Software.
echo.

:: Obtain Ventoy (Only 32-bit Package File)

echo Starting to %script_execution_action% Ventoy ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=Ventoy (X%%a)"
    set "id_package=Ventoy.Ventoy -a x%%a"
    set "destination_folder=Burn_and_Format_Software\Ventoy"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Ventoy X%%a"
    set "newname_packagefile=Ventoy_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=zip"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 7: Virtualization Software.
echo.

:: Obtain VirtualBox (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% VirtualBox ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=VirtualBox (X%%a)"
    set "id_package=Oracle.VirtualBox -a x%%a"
    set "destination_folder=Virtualization_Software\VirtualBox"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=VirtualBox X%%a"
    set "newname_packagefile=VirtualBox_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1

)
```

```
echo.
echo Software Pack 8: Maintain Software.
echo.

:: Obtain TeamViewer (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% TeamViewer ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=TeamViewer (X%%a)"
    set "id_package=TeamViewer.TeamViewer -a x%%a"
    set "destination_folder=Maintain_Software\TeamViewer"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=TeamViewer X%%a"
    set "newname_packagefile=TeamViewer_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain AnyDesk (Only 32-bit Package File)

echo Starting to %script_execution_action% AnyDesk ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=AnyDesk (X%%a)"
    set "id_package=AnyDeskSoftwareGmbH.AnyDesk -a x%%a"
    set "destination_folder=Maintain_Software\AnyDesk"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=AnyDesk X%%a"
    set "newname_packagefile=AnyDesk_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 9: Network Software.
echo.

:: Obtain Google Chrome (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Google Chrome ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=Google Chrome (X%%a)"
    set "id_package=Google.Chrome -a x%%a"
    set "destination_folder=Network_Software\Google_Chrome"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Google Chrome (X%%a)"
    set "newname_packagefile=Google_Chrome_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=msi"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Opera (Only 64-bit Package File)

echo Starting to %script_execution_action% Opera ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=Opera (X%%a)"
    set "id_package=Opera.Opera -a x%%a"
    set "destination_folder=Network_Software\Opera"
```

```
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Opera (X%%a)"
    set "newname_packagefile=Opera_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Google Drive (Only 64-bit Package File)

echo Starting to %script_execution_action% Google Drive ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=Google Drive (X%%a)"
    set "id_package=Google.GoogleDrive -a x%%a"
    set "destination_folder=Network_Software\Google_Drive"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Google Drive (X%%a)"
    set "newname_packagefile=Google_Drive_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Bitwarden (Only 32-bit Package File)

echo Starting to %script_execution_action% Bitwarden ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=Bitwarden (X%%a)"
    set "id_package=Bitwarden.Bitwarden"
    set "destination_folder=Network_Software\Bitwarden"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Bitwarden (%%a)"
    set "newname_packagefile=Bitwarden_%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain qBittorrent (Only 64-bit Package File)

echo Starting to %script_execution_action% qBittorrent ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=qBittorrent (X%%a)"
    set "id_package=qBittorrent.qBittorrent -a x%%a"
    set "destination_folder=Network_Software\qBittorrent"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=qBittorrent (X%%a)"
    set "newname_packagefile=qBittorrent_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 10: Editing and Capturing Software.
echo.

:: Obtain GIMP (32-bit and 64-bit Package Files)
```

```
echo Starting to %script_execution_action% GIMP ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=GIMP (X%%a)"
    set "id_package=GIMP.GIMP -a x%%a"
    set "destination_folder=Editing_and_Capturing_Software\GIMP"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=GIMP (X%%a)"
    set "newname_packagefile=GIMP_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Paint.Net (Only 64-bit Package File)

echo Starting to %script_execution_action% Paint.Net ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=Paint.Net (X%%a)"
    set "id_package=dotPDN.PaintDotNet -a x%%a"
    set "destination_folder=Editing_and_Capturing_Software\Paint.Net"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Paint Net (X%%a)"
    set "newname_packagefile=Paint.Net_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=zip"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain OBSStudio (Only 64-bit Package File)

echo Starting to %script_execution_action% OBSStudio ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=OBSStudio (X%%a)"
    set "id_package=OBSProject.OBSStudio -a x%%a"
    set "destination_folder=Editing_and_Capturing_Software\OBSStudio"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=OBSStudio (X%%a)"
    set "newname_packagefile=OBSStudio_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 11: Multimedia Software.
echo.

:: Obtain VLC (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% VLC ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=VLC (X%%a)"
    set "id_package=VideoLAN.VLC -a x%%a"
    set "destination_folder=Multimedia_Software\VLC"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=VLC (X%%a)"
    set "newname_packagefile=VLC_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
```

```
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain K-Lite Codec Pack Mega (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% K-Lite Codec Pack Mega ^(32-bit/64-
bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=K-Lite Codec Pack Mega (X%%a)"
    set "id_package=CodecGuide.K-LiteCodecPack.Mega -a x%%a"
    set "destination_folder=Multimedia_Software\K-Lite_Codec_Pack_Mega"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=K-Lite Codec Pack Mega (X%%a)"
    set "newname_packagefile=K-Lite_Codec_Pack_Mega_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Winamp (Only 32-bit Package File)

echo Starting to %script_execution_action% Winamp ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=Winamp (X%%a)"
    set "id_package=Winamp.Winamp -a x%%a"
    set "destination_folder=Multimedia_Software\Winamp"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Winamp (X%%a)"
    set "newname_packagefile=Winamp_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain FastStone Viewer (Only 32-bit Package File)

echo Starting to %script_execution_action% FastStone Viewer ^(Only 32-bit^) &
echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=FastStone Viewer (X%%a)"
    set "id_package=FastStone.Viewer -a x%%a"
    set "destination_folder=Multimedia_Software\FastStone_Viewer"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=FastStone Viewer (X%%a)"
    set "newname_packagefile=FastStone_Viewer_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

echo.
echo Software Pack 12: Communication Software.
echo.

:: Obtain Zoom (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Zoom ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=Zoom (X%%a)"
    set "id_package=Zoom.Zoom -a x%%a"
```

```
    set "destination_folder=Communication_Software\Zoom"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Zoom (X%%a)"
    set "newname_packagefile=Zoom_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=msi"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Microsoft Skype (Only 32-bit Package File)

echo Starting to %script_execution_action% Microsoft Skype ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=Microsoft Skype (X%%a)"
    set "id_package=Microsoft.Skype -a x%%a"
    set "destination_folder=Communication_Software\Microsoft_Skype"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Microsoft Skype (X%%a)"
    set "newname_packagefile=Microsoft_Skype_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Discord (Only 64-bit Package File)

echo Starting to %script_execution_action% Discord ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=Discord (X%%a)"
    set "id_package=Discord.Discord -a x%%a"
    set "destination_folder=Communication_Software\Discord"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Discord (X%%a)"
    set "newname_packagefile=Discord_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Telegram Desktop (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Telegram Desktop ^(32-bit/64-bit^) &
echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=Telegram Desktop (X%%a)"
    set "id_package=Telegram.TelegramDesktop -a x%%a"
    set "destination_folder=Communication_Software\Telegram_Desktop"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Telegram Desktop (X%%a)"
    set "newname_packagefile=Telegram_Desktop_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Viber Only With "--accept-package-agreements" to Accept All License
Agreements For Packages Silenty (Only 64-bit Package File)

echo Starting to %script_execution_action% Viber ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=Viber (X%%a)"
```

```
    set "id_package=XPFM5P5KDWF0JP -a x%%a --accept-package-agreements"
    set "destination_folder=Communication_Software\Viber"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Viber (X%%a)"
    set "newname_packagefile=Viber_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=msi"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

echo.
echo Software Pack 13: Playground Software.
echo.

:: Obtain Steam (Only 32-bit Package File)

echo Starting to %script_execution_action% Steam ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=Steam (X%%a)"
    set "id_package=Valve.Steam -a x%%a"
    set "destination_folder=Playground_Software\Steam"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Steam (X%%a)"
    set "newname_packagefile=Steam_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Epic Games Launcher (Only 32-bit Package File)

echo Starting to %script_execution_action% Epic Games Launcher ^(Only 32-bit^) &
echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=Epic Games Launcher (X%%a)"
    set "id_package=EpicGames.EpicGamesLauncher -a x%%a"
    set "destination_folder=Playground_Software\Epic_Games_Launcher"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Epic Games Launcher (X%%a)"
    set "newname_packagefile=Epic_Games_Launcher_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=msi"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 14: Developing Software.
echo.

:: Obtain GitHub Desktop (Only 64-bit Package File)

echo Starting to %script_execution_action% GitHub Desktop ^(Only 64-bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=GitHub Desktop (X%%a)"
    set "id_package=GitHub.GitHubDesktop -a x%%a"
    set "destination_folder=Developing_Software\GitHub_Desktop"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=GitHub Desktop (X%%a)"
    set "newname_packagefile=GitHub_Desktop_X%%a"
```

```
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Microsoft Visual Studio Code (Only 64-bit Package File)

echo Starting to %script_execution_action% Microsoft Visual Studio Code ^(Only 64-
bit^) & echo.
for %%a in (%cpu_arch_64%) do (
    set "name_package=Microsoft Visual Studio Code (X%%a)"
    set "id_package=Microsoft.VisualStudioCode -a x%%a"
    set "destination_folder=Developing_Software\Microsoft_Visual_Studio_Code"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Microsoft Visual Studio Code (X%%a)"
    set "newname_packagefile=Microsoft_Visual_Studio_Code_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Python 3.12 (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Python 3.12 ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=Python 3_12 (X%%a)"
    set "id_package=Python.Python.3.12"
    set "destination_folder=Developing_Software\Python_3.12"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Python 3.12 (X%%a)"
    set "newname_packagefile=Python_3.12_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)


echo.
echo Software Pack 15: Office Software.
echo.

:: Obtain Foxit Reader (Only 32-bit Package File)

echo Starting to %script_execution_action% Foxit Reader ^(Only 32-bit^) & echo.
for %%a in (%cpu_arch_32%) do (
    set "name_package=Foxit Reader (X%%a)"
    set "id_package=Foxit.FoxitReader -a x%%a"
    set "destination_folder=Office_Software\Foxit_Reader"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Foxit Reader (X%%a)"
    set "newname_packagefile=Foxit_Reader_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Obtain Notepad++ (32-bit and 64-bit Package Files)

echo Starting to %script_execution_action% Notepad++ ^(32-bit/64-bit^) & echo.
for %%a in (%cpu_arch_current%) do (
    set "name_package=Notepad++ (X%%a)"
```

```
    set "id_package=Notepad++.Notepad++ -a x%%a"
    set "destination_folder=Office_Software\Notepad++"
    call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
    set "secondname_packagefile=Notepad++ (X%%a)"
    set "newname_packagefile=Notepad++_X%%a"
    set "arch_packagefile=%%a"
    set "type_packagefile=exe"
    call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
)

:: Script Completion Message

echo Everything is %script_execution_action%ed Successfully. & echo.

if "%script_execution_action%"=="Download" (
    echo Your Offline Package is Ready to Use. & echo. & echo.
) else if "%script_execution_action%" =="Install" (
    echo The Software is Ready to Use. & echo. & echo.
)

call :SUCCESS_END
goto :eof


:SAMPLE_PACK

echo Software Pack 1: System Software & echo.
echo    [01] - Microsoft Visual C++ Runtimes
echo    [02] - Microsoft .NET Runtimes
echo    [03] - Microsoft DirectX
echo    [04] - Java 8 Runtime Envirovment
echo    [05] - PowerShell 7 & echo.

echo Software Pack 2: File Archiver Software & echo.
echo    [01] - WinRAR & echo.

echo Software Pack 3: System Monitoring Software & echo.
echo    [01] - AIDA64 Engineer
echo    [02] - HWiNFO
echo    [03] - CrystalDiskInfo
echo    [04] - WinDirStat & echo.

echo Software Pack 4: Testing Software & echo.
echo    [01] - CrystalDiskMark & echo.

echo Software Pack 5: Backup and Managment Software & echo.
echo    [01] - Recuva & echo.

echo Software Pack 6: Burn and Format Software & echo.
echo    [01] - Ventoy
echo    [02] - Ultraiso & echo.

echo Software Pack 7: Virtualization Software & echo.
echo    [01] - VirtualBox & echo.

echo Software Pack 8: Maintain Software & echo.
echo    [01] - TeamViewer
echo    [02] - AnyDesk & echo.
```

```
echo Software Pack 9: Network Software & echo.
echo    [01] - Google Chrome
echo    [02] - Opera
echo    [03] - Google Drive
echo    [04] - Bitwarden
echo    [05] - qBittorrent & echo.

echo Software Pack 10: Editing and Capturing Software & echo.
echo    [01] - GIMP
echo    [02] - Paint.Net
echo    [04] - OBSStudio & echo.

echo Software Pack 11: Multimedia Software & echo.
echo    [01] - VLC
echo    [02] - K-Lite Codec Pack Mega
echo    [03] - Winamp
echo    [04] - FastStone Viewer & echo.

echo Software Pack 12: Communication Software & echo.
echo    [01] - Zoom
echo    [02] - Microsoft Skype
echo    [03] - Discord
echo    [04] - Telegram Desktop
echo    [05] - Viber & echo.

echo Software Pack 13: Playground Software & echo.
echo    [01] - Steam
echo    [02] - Epic Games Launcher & echo.

echo Software Pack 14: Developing Software & echo.
echo    [01] - GitHub Desktop
echo    [03] - Microsoft Visual Studio Code
echo    [04] - Python 3.12 & echo.

echo Software Pack 15: Office Software & echo.
echo    [01] - Foxit Reader
echo    [02] - Notepad++ & echo.
goto :eof

:OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE

winget %script_execution_action% !id_package! -d
C:\TEMP\Offline_Pack\!destination_folder!\ & echo.
if !errorlevel! equ 0 (
    echo !name_package! has been %script_execution_action%ed Successfully. & echo.
& echo.
) else (
    echo !name_package! Failed to %script_execution_action%. & echo. & echo.
)
goto :eof


:RENAME_PACKAGEFILE_WITH_CHECK_CYCLE

cd /d "C:\TEMP\Offline_Pack\!destination_folder!" > nul 2>&1
del "C:\TEMP\Offline_Pack\!destination_folder!\*.yaml" /s /q > nul 2>&1
for /f "delims=" %%i in ('dir /b "*!arch_packagefile!*.!type_packagefile!"') do (
    ren "%%i" "!newname_packagefile!.!type_packagefile!"
    if !errorlevel! equ 0 (
```

```
        echo !secondname_packagefile! Successfully Renamed to
!newname_packagefile!.!type_packagefile!.
    ) else (
        echo !secondname_packagefile! Not Found.
    )
)
goto :eof


:ERROR_END
echo.
echo Please Restart the Script or Try Again Later.
goto :REAL_END

:SUCCESS_END

echo.
echo Done.
goto :REAL_END

:REAL_END

echo.
pause
exit
```

Додаток Г

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістра

НА ТЕМУ: «СИСТЕМА АВТОМАТИЗАЦІЇ ОБСЛУГОВУВАННЯ ПРИСТРОЇВ В УМОВАХ ВІДДАЛЕНОЇ РОБОТИ»

Виконав студент групи КСДМ-62

Гафанович Владислав Олександрович

Керівник дипломної роботи:

Асєєва Людмила Анатоліївна,

phD (доктор філософії), доцент

Київ – 2025

Мета, об’єкт, предмет дослідження 2

Мета роботи – розробити та дослідити систему автоматизації обслуговування пристроїв для забезпечення ефективного управління та моніторингу в умовах віддаленої роботи, з акцентом на зниження витрат і підвищення надійності

Об’єкт дослідження – процес обслуговування та управління пристроями на віддалених робочих місцях

Предмет дослідження – засоби та методи автоматизації обслуговування пристроїв в умовах віддаленої роботи

Актуальність роботи

3

У роботі аналізуються основні аспекти автоматизації обслуговування пристроїв для віддалених працівників, зокрема роль хмарних рішень, автоматизованих оновлень і діагностики. Окрему увагу приділено питанням безпеки, автономності та масштабованості таких систем, що дозволяє ефективно управляти інфраструктурою без постійної залежності від інтернет-з’єднання.

Запропоновані підходи оптимізують процеси обслуговування за допомогою скриптів і автоматичних оновлень, що мінімізують втручання операторів і знижують витрати на підтримку пристроїв. Розроблена система також демонструє економічний ефект завдяки скороченню витрат на ручне обслуговування і зниженню потреби в додаткових ресурсах.

Проблема та необхідність автоматизації обслуговування пристроїв у віддаленій роботі

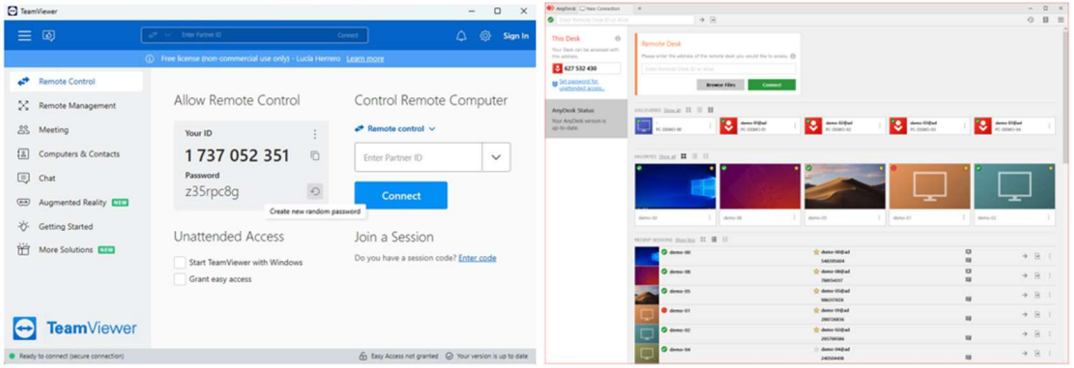

Рис. 1.1 – ПЗ для віддаленої підтримки TeamViewer та AnyDesk (не адаптоване для підтримки великої кількості пристроїв; в разі використання у комерційних цілях – блокування)

Рис. 1.2 – Різноманіття кінцевих пристроїв у розподілених системах (пристрої, що використовують віддалені працівники)

Рис. 1.3 – Складність мережевої інфраструктури під час віддаленої роботи

5

Процес впровадження автоматизації у віддаленому обслуговуванні

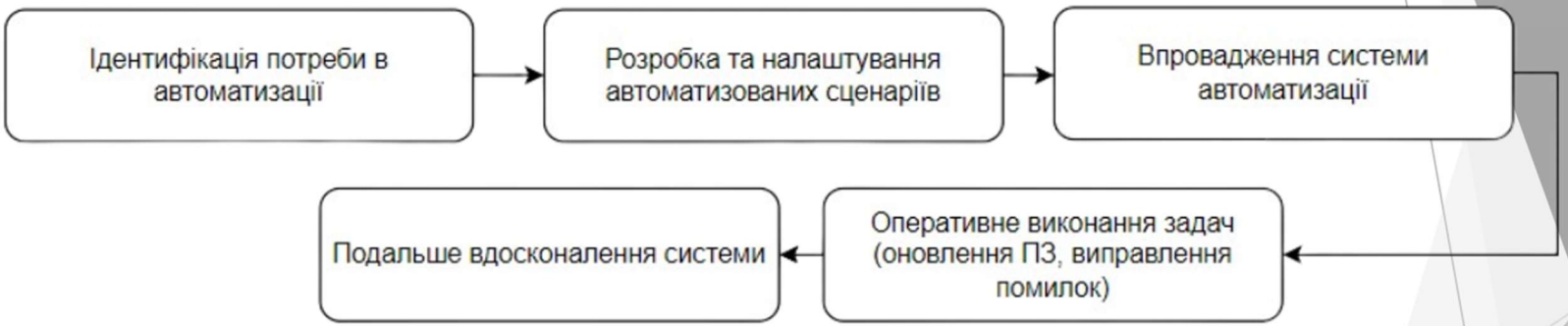

Рис. 1.11 – Етапи впровадження автоматизації в процес віддаленого обслуговування

6 Автоматизація обслуговування через сценарії (скрипти)

Рис. 3.20 – Блок-схема роботи сценарію безперервного розгортання ПЗ

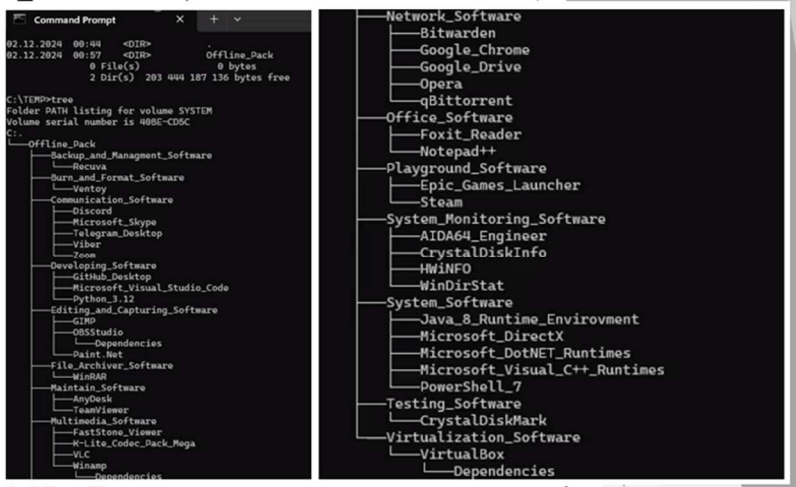

Рис. 3.27 – Дерево папки, куди завантажуються усі пакети програм (для встановлення чи збереження на ПК, звідки був запущений скрипт)

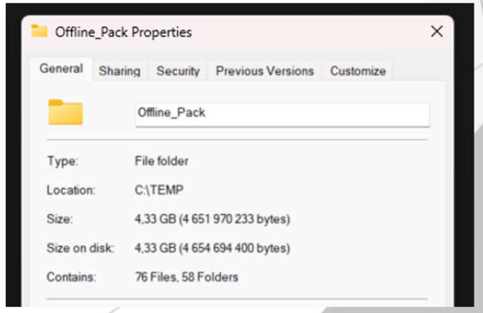

Рис. 3.28 – Розмір автономного пакета програм, що сформувався у папці TEMP

Технічні вимоги та залежності для сценаріїв (скриптів)

7

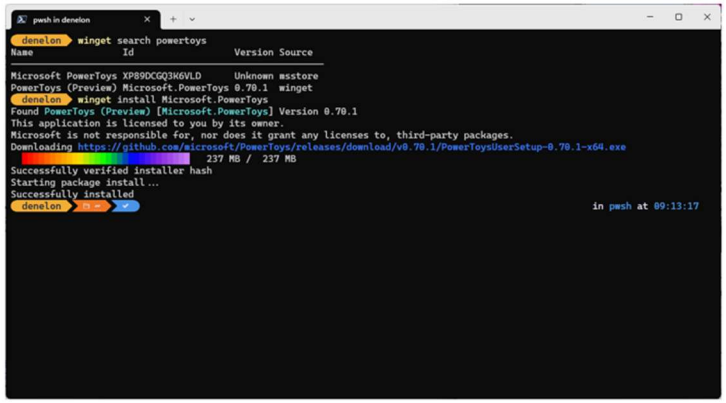

Рис. 2.29 – Пакетний менеджер Winget (Windows Package Manager)

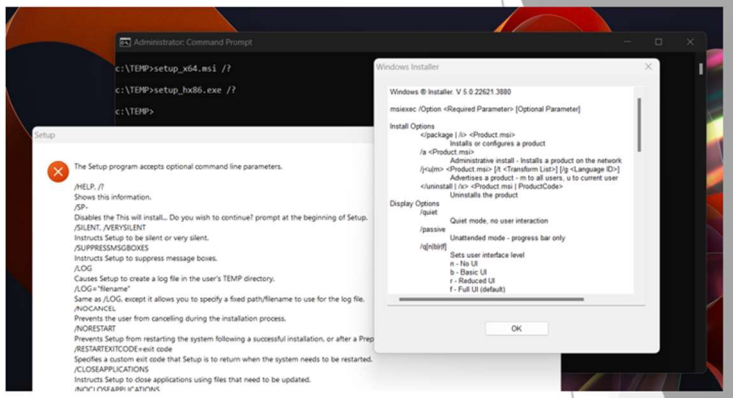

Рис. 3.9 – Наявність у інсталлерів параметрів для швидкої непомітної розгортки ПЗ

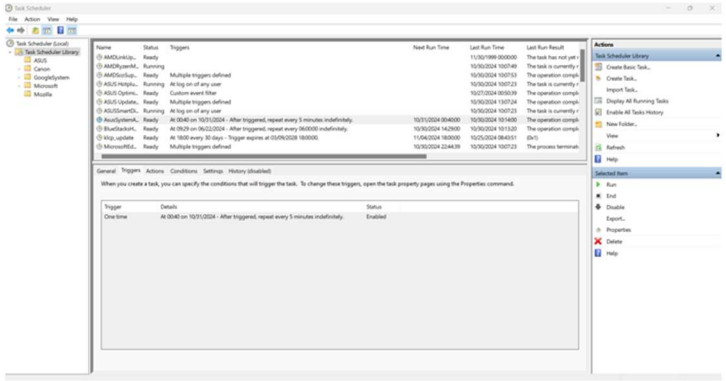

Рис. 3.10 – Використання Task Scheduler як заплановане виконання сценаріїв (скриптів) автоматизації

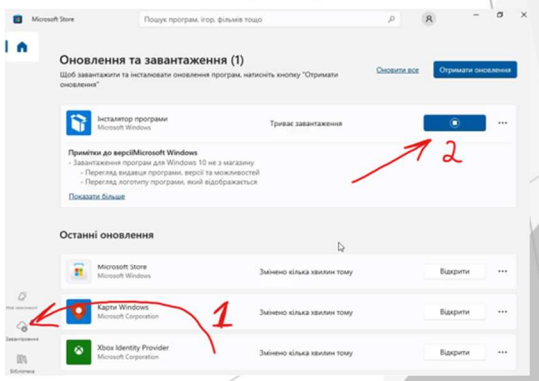

Рис. 3.14 – Оновлення Windows Installer через Microsoft Store (для отримання Winget)

Покрокова демонстрація роботи сценарію (скрипта) безперервного розгортання ПЗ

8

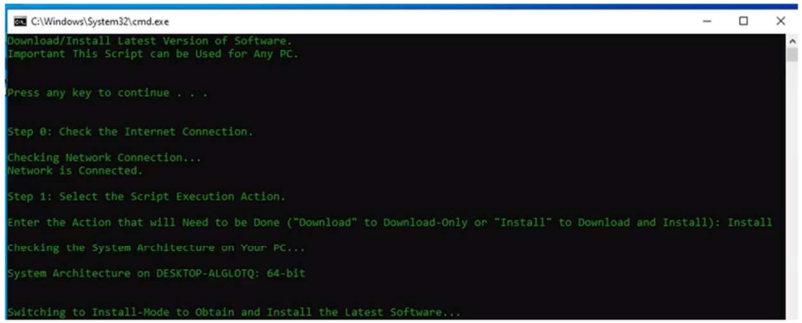

Рис. 3.23 – Вибір дії сценарію виконання
(лише завантажити пакет ПЗ або завантажити та встановити пакет ПЗ)

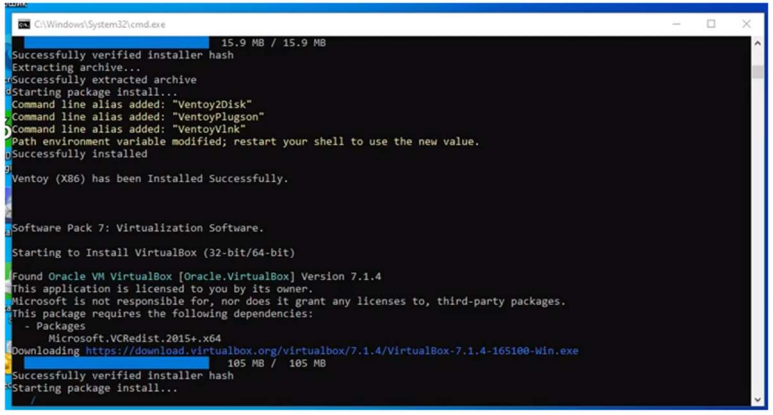

Рис. 3.30 – Процес завантаження або/та встановлення пакету ПЗ

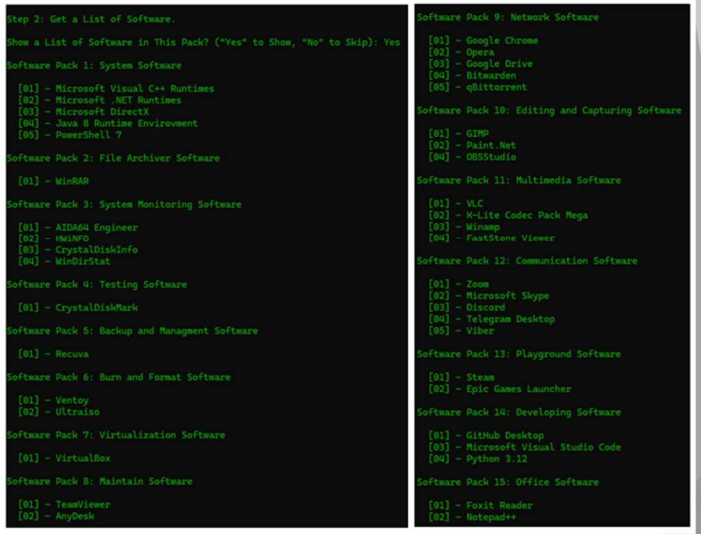

Рис. 3.25 – Список пакету ПЗ, до якого входить 15 підпакетів, поділених на категорії

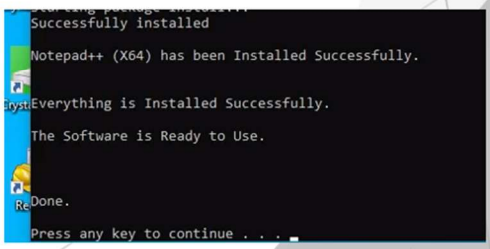

Рис. 3.34 – Сценарій успішно виконав «тіло» сценарію

Фрагмент скрипта та виклики підпрограм


```
533
534   :: Obtain K-Lite Codec Pack Mega (32-bit and 64-bit Package Files)
535
536   echo Starting to %script_execution_action% K-Lite Codec Pack Mega ^(32-bit/64-bit^) & echo.
537   for %%a in (%cpu_arch_current%) do (
538       set "name_package=K-Lite Codec Pack Mega (X%%a)"
539       set "id_package=CodecGuide.K-LiteCodecPack.Mega -a x%%a"
540       set "destination_folder=Multimedia_Software\K-Lite_Codec_Pack_Mega"
541       call :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
542       set "secondname_packagefile=K-Lite Codec Pack Mega (X%%a)"
543       set "newname_packagefile=K-Lite_Codec_Pack_Mega_X%%a"
544       set "arch_packagefile=%%a"
545       set "type_packagefile=exe"
546       call :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE > nul 2>&1
547   )
548
```

Рис. 3.32 – Перед зверненням до підпрограми відбувається присвоєння змінним нових значень

```
863   goto :eof
864
865   :OBTAIN_LATEST_SOFTWARE_WITH_CHECK_CYCLE
866
867   winget %script_execution_action% !id_package! -d C:\TEMP\Offline_Pack\!destination_folder!\ & echo.  !!
868   if !errorlevel! equ 0 (
869       echo !name_package! has been %script_execution_action%ed Successfully. & echo. & echo.
870   ) else (
871       echo !name_package! Failed to %script_execution_action%. & echo. & echo.
872   )
873   goto :eof
874
875
876   :RENAME_PACKAGEFILE_WITH_CHECK_CYCLE
877
878   cd /d "C:\TEMP\Offline_Pack\!destination_folder!" > nul 2>&1
879   del "C:\TEMP\Offline_Pack\!destination_folder!\*.yaml" /s /q > nul 2>&1
880   for /f "delims=" %%i in ('dir /b "*!arch_packagefile!*.!type_packagefile!"') do (
881       ren "%%i" "!newname_packagefile!.!type_packagefile!"
882       if !errorlevel! equ 0 (
883           echo !secondname_packagefile! Successfully Renamed to !newname_packagefile!.!type_packagefile!.
884       ) else (
885           echo !secondname_packagefile! Not Found.
886       )
887   )
888   goto :eof
889
```

Рис. 3.31 – Звернення до підпрограм задля зменшення розмірності сценарію (головні підпрограми, що виконують завантаження або/та встановлення пакету програм, створюючи при цьому структуру за вказаним патерном)

```
889
890
891   :ERROR_END
892   echo.
893   echo Please Restart the Script or Try Again Later.
894   goto :REAL_END
895
896   :SUCCESS_END
897
898   echo.
899   echo Done.
900   goto :REAL_END
901
902   :REAL_END
903
904   echo.
905   pause
906   exit
```

Рис. 3.33 – Варіанти завершення роботи сценарію (невірно введено відповідь; успішне завершення після виконання «тіла» скрипта)

Отримані результати від впровадження автоматизованого рішення

10

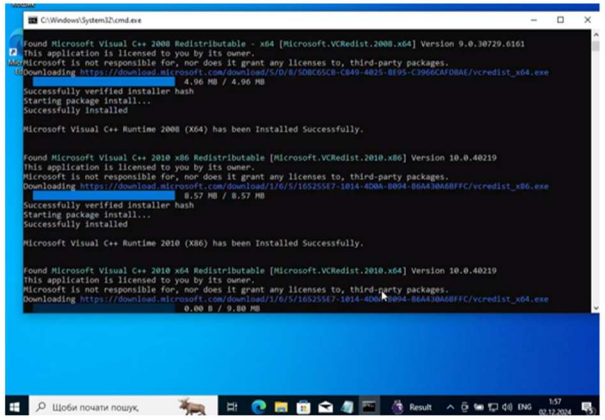

Рис. 3.36 – Початок роботи сценарію безперервоного розгортання ПЗ (дата та час початку: 02.12.2024 – 1:57)

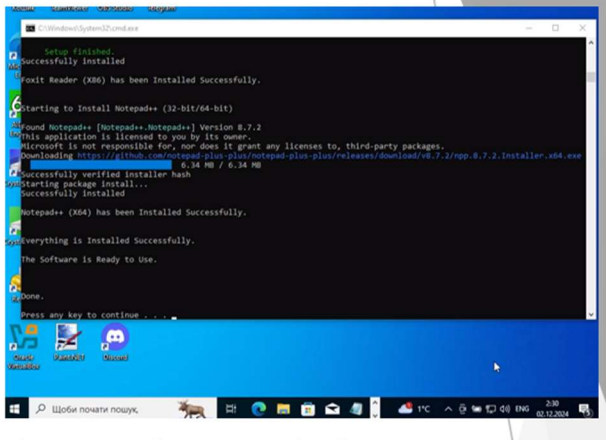

Рис. 3.37 – Закінчення роботи сценарію безперервоного розгортання ПЗ (дата та час закінчення: 02.12.2024 – 2:30)

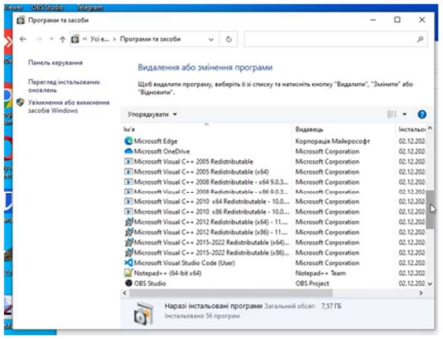

Рис. 3.35 – Пакет ПЗ був успішно розвернутий на новій ОС (встановлено 42 програми)

Рис. 3.38 – Відображено робочий стіл зі розгорнутим пакетом ПЗ

Порівняння запропонованого рішення

11

Критерій	**Запропоноване рішення (автоматизований сценарій)**	Комерційні-рішення (Microsoft Intune, Entra тощо)
Вартість	**Початкові інвестиції, відсутність ліцензійних платежів.**	Постійні витрати на ліцензії, що залежать від кількості користувачів чи пристроїв.
Гнучкість налаштувань	**Максимальна адаптивність, налаштування під конкретні потреби організації.**	Обмежена функціональність, стандартизовані підходи.
Автономність	**Працює локально, не залежить від постійного інтернет-з'єднання чи хмарних сервісів. Інтернет використовується лише для завантаження необхідних пакетів ПЗ.**	Залежність від хмарних серверів та інтернет-з'єднання.
Швидкість розгортання	**Встановлення 42 програм за 23 хвилини. Ручний процес зайняв би понад годину.**	Час залежить від хмарних серверів та доступності інтернету.
Масштабованість	**Легке масштабування без додаткових витрат.**	Додаткові витрати на масштабування та зміну тарифних планів.
Захист даних	**Локальна обробка, що зменшує ризики витоку інформації.**	Дані зберігаються в хмарі, ризик витоків чи залежність від політик постачальника.
Залежність від постачальника	**Повна незалежність, використання відкритих технологій.**	Висока залежність від конкретного постачальника і його екосистеми.
Оптимізація витрат на ПЗ	**Автоматична аналітика для управління ліцензіями, відмова від зайвих програм.**	Можливості оптимізації обмежені стандартними звітами.
Контроль оновлень	**Локальне тестування оновлень, мінімізація ризиків простоїв.**	Автоматичне розгортання, але ризики конфліктів залишаються.

Табл. 3.1 – Порівняння запропонованого рішення між комерційними

12 Висновки

Автоматизація технічного обслуговування кінцевих пристроїв у сучасному бізнес-середовищі відкриває нові можливості для ефективного управління ресурсами та оптимізації процесів. Пропоноване рішення забезпечує швидке розгортання ПЗ, демонструючи значні переваги у швидкості виконання завдань. Зокрема, можливість встановлювати 42 програми за 23 хвилини, що значно перевершує понад годину, необхідну для ручного процесу, робить це рішення максимально практичним і економічно вигідним.

Ключовою перевагою є автономність, що дозволяє системі працювати без постійного підключення до інтернету. Завантаження необхідних пакетів ПЗ відбувається лише один раз, що мінімізує залежність від зовнішніх факторів, таких як мережеві збої чи нестабільне інтернет-з’єднання, що забезпечує високу надійність навіть у віддалених або малодоступних місцях, де доступ до мережі є обмеженим.

Продовження висновків

13

Продуктивність і безпека системи підвищуються завдяки централізованому керуванню пристроями, швидкому оновленню програмного забезпечення та зниженню ручної праці. Такий підхід мінімізує ризик людських помилок, скорочує час простоїв і захищає інфраструктуру від кіберзагроз завдяки оперативному впровадженню патчів безпеки. Масштабованість рішення дозволяє ефективно інтегрувати нові пристрої та користувачів, зменшуючи витрати на додатковий персонал.

Потенціал автоматизації може зрости у майбутньому зі впровадженням сучасних технологій, таких як штучний інтелект і машинне навчання, які б дозволили прогнозувати можливі збої, запобігати їм завчасно та автоматизувати управління всією інфраструктурою. Таким чином, система стає ключовим елементом у цифровій трансформації бізнесу, відповідаючи вимогам як сьогодення, так і майбутнього розвитку

Публікації та апробація роботи 14

1. Гафанович В.О., Антоненко А.В. Впровадження системи електронного документообігу та управління даними в корпоративній мережі.«Застосування програмного забезпечення в ІКТ» : Всеукр. науково-техн. конф., м.Київ, 24 квітня 2024 р. С. 180-182. URL: https://duikt.edu.ua/uploads/p_2661_45497999.pdf

2. BUILDING CORPORATE NETWORKS FOR FOOD INDUSTRY ENTERPRISES / A. BALVAK, M. DEMCHENKO, V. DRAHUN, V. PETRENKO, V. PRYSIAZHNIUK, O. IVANOV, S. KOROTKOV, O. TSVYK, O. TONKYKH, A. ANTONENKO, A. LEMESHKO, V. HAFANOVYCH, O. HOLUBENKO. European Science, (sge22-01), (2023), P.25–47. URL: https://doi.org/10.30890/2709-2313.2023-22-01-031

3. Гафанович В.О., Асеєва Л.А. Автоматизація управління програмним забезпеченням на Windows за допомогою Winget. «Проблеми комп’ютерної інженерії» : Науково-практ. конф., м.Київ, 3 грудня 2024 р. С.181-183

4. Гафанович В.О., Асеєва Л.А. Використання сценарія для встановлення програмного забезпечення онлайн та офлайн режимах. «Проблеми комп’ютерної інженерії» : Науково-практ. конф., м.Київ, 3 грудня 2024 р. С.184-185