

# ГЕОІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ МОДЕЛЮВАННЯ СТАНУ ГЕОСИСТЕМ

ISSN 2306-1677 (Print) ISSN 2518-7325 (On-line)

Отримано: 06.05.2026; Прийнято: 02.06.2026; Опубліковано: 30.06.2026;;

УДК 004.424.22:528.235

<http://dx.doi.org/10.31548/zemleustriy2026.02.01>

## ЗАСТОСУВАННЯ МЕТОДУ ВКЛАДЕНОЇ ІТЕРАЦІЇ ДЛЯ АВТОМАТИЗОВАНОГО КАРТОГРАФУВАННЯ

**Й.М. ДОРОШ,**

*доктор економічних наук, професор, академік НААН України,*

*E-mail: doroshjosyp@gmail.com*

*RCID: 0000-0002-1764-6188*

*Інститут землекористування НААН України*

**Б.Б. ЗАЯЧКІВСЬКА,**

*кандидат економічних наук,*

*E-mail: b\_zayachkivska@nubip.edu.ua*

*ORCID: 0009-0003-0962-5998*

*Національний університет біоресурсів і природокористування України*

**Р.А. ХАРИТОНЕНКО,**

*кандидат економічних наук,*

*E-mail: kharytonenkor@gmail.com*

*ORCID: 0000-0002-7992-3339*

*Інститут землекористування НААН України*

**Д.М. МЕЛЬНИК,**

*кандидат економічних наук,*

*E-mail: melnykdenys@gmail.com*

*ORCID: 0000-0002-3714-9592*

*Інститут землекористування НААН України*

**Анотація.** У статті розглядається метод автоматизації створення серії картографічних матеріалів на основі інтеграції інструментарію QGIS Atlas та мови програмування Python. Метою дослідження є оптимізація процесу підготовки багатосторінкових тематичних атласів для великих масивів просторових даних. Для досягнення поставленої мети застосовано авторський підхід до інтеграції алгоритмів вкладеної ітерації, що дозволяє гнучко адаптувати просторові межі та індивідуальні параметри візуального відображення для

кожного елемента системи. Ключовим елементом є розроблений алгоритм, що базується на обробці події `featureChanged`, що забезпечує автоматичну синхронізацію атрибутів району з відповідними файлами стилів `.qml`. Модель апробовано на масиві даних обсягом понад 20 тис. контурів. Результати апробації підтвердили високу ефективність методу. Швидкість генерації серії карт зростає у приблизно 20 разів порівняно з традиційними методами, при повному виключенні суб'єктивних помилок візуалізації. Запропонований підхід становить собою універсальну методологічну платформу для автоматизованої трансформації складних просторових датасетів у високоточні графічні моделі. Дане рішення забезпечує можливість оперативного представлення результатів багатофакторних досліджень у будь-якій галузі, де вимагається поєднання високої інтенсивності обробки великих масивів інформації з дотриманням індивідуальних параметрів візуальної інтерпретації для кожного окремого елемента системи.

**Ключові слова:** автоматизоване картографування, QGIS Atlas, PyQGIS, агрохімічний моніторинг, ГІС-автоматизація, метод вкладеної ітерації, динамічна стилізація, тематичні атласи, Python API, файли стилів `*.qml`, автоматизація макетів, алгоритмізація картографування.

---

### Актуальність

Тематичне картографування є обов'язковим інструментом для забезпечення потреб державного управління, оборони, територіального планування та моніторингу навколишнього природного середовища [1] та передбачає відображення об'єктів і явищ у конкретних галузях (економіка, екологія, сільське господарство тощо), що потребує високої точності та регулярного оновлення даних. Традиційне створення тематичних карт для великої кількості адміністративних одиниць (районів, громад, населених пунктів, сільськогосподарських полів) є рутинним процесом, що потребує значних часових витрат. Кожна карта вимагає індивідуального налаштування фільтрації шарів, зумування та застосування відповідної шкали кольорів.

QGIS з відкритим кодом (Open Source) на сьогодні є однією з найпопулярніших геоінформаційних систем

завдяки можливостям безкоштовного використання і постійних оновленням [2]. Дана програма широко застосовується в землевпорядкуванні, агрономії, лісівництві, управлінні водними ресурсами, екології, містобудуванні та державному управлінні для обробки великих масивів просторових даних. Завдяки кросплатформеності та підтримці мови високого рівня програмування Python, QGIS став популярним інструментом для створення складних картографічних звітів, не тільки для візуалізації, але і автоматизації циклічних завдань у моніторингу земельних ресурсів.

Вибір програмного середовища QGIS для реалізації методу вкладеної ітерації базується на висновках S. Steiniger та A. Hunter щодо функціональної спроможності вільного ПЗ. Автори зазначають, що сучасні відкриті ГІС-платформи забезпечують необхідну гнучкість для вирішення специфічних задач геообробки, що дозволяє інтегрувати складні алгоритми автоматизації без значних ка-

піталовкладень [3]. Необхідність автоматизації картографічного процесу підтверджується сучасними тенденціями у цифровому ґрунтознавстві (Digital Soil Mapping) [4]. Застосування інтелектуальних моделей машинного навчання для прогнозування просторової мінливості характеристик ґрунтів, таких як вміст органічного вуглецю та рівень рН, вимагає адекватних інструментів динамічної візуалізації [5]. На думку Р. Еспінеля та співавторів використання штучного інтелекту та автоматизованих алгоритмів дозволяє суттєво скоротити часові витрати на обробку просторових даних, забезпечуючи при цьому високу ефективність у прийнятті управлінських рішень [6].

Впровадження алгоритмів автоматизованої генерації карт забезпечує застосування визначених шаблонів, надає можливість створення тисяч тематичних планів за лічені години, що фізично неможливо при ручному компонуванні та підвищує якість прийняття рішень так як установи можуть отримувати актуальні атласи в режимі реального часу, що дозволяє миттєво реагувати на негативні зміни в стані земельних ресурсів. Тому використання ГІС-інструментарію в землеустрої України є не просто технічним покращенням, а фундаментальним механізмом трансформації всієї галузі [7], а розвиток ГІС-технологій дозволяє вітчизняним установам відповідати міжнародним стандартам управління даними та забезпечувати прозорість моніторингу територій.

### ***Аналіз останніх досліджень та публікацій***

Б. Сінгх підтверджує, що ГІС використовується міністерствами

(наприклад, USDA) для захисту врожаю, розв'язання економічних питань та дослідження стану ґрунтів. Сільськогосподарські ГІС здатні трансформувати величезні обсяги даних у наочні карти для підтримки прийняття рішень, що робить QGIS незамінним інструментом у моніторингу стану ґрунтового покриву і сільськогосподарських посівів завдяки його технічним можливостям та доступності. Сінг оцінює високу точність даних на рівні 90-го процентиля, однак процес візуалізації таких масштабних масивів залишається трудомістким, що зумовлює необхідність впровадження алгоритмів автоматизованого динамічного картографування [8].

Більшість існуючих робіт присвячені або QGIS Atlas без Python, або повній генерації карт через Python без використання інтерфейсу QGIS. Так, Аніта Гразер описує автоматизацію через використання макетів та атласу як фундаментальну складову серійного картографування, але вказує на його обмеження, так як програмний модуль сам по собі не передбачає змінювання файлів стилів (.qml) для кожного об'єкта [9].

За словами Г. Шермана, поєднання QGIS та Python (PyQGIS) дозволяє розширювати інструментарій ГІС з відкритим кодом, створюючи скрипти для виконання складних автоматизованих завдань, що неможливо реалізувати стандартними засобами інтерфейсу. Автор розглядає Python як основний інструмент для написання плагінів та розширення API [10]. Отже, більшість прикладів автоматизації стосуються обробки даних у консолі, тоді як метод вкладеної ітерації інтегрує Python-скрипти безпосередньо в цикл роботи серійного

компонувальника макетів (Atlas) в один результуючий файл.

Г. Шерман обґрунтовує концепцію розширення базового інструментарію ГІС шляхом інтеграції Python 3 з QGIS API для виконання складних автоматизованих завдань. В основу методу вкладеної ітерації покладено принцип ієрархічного управління об'єктами проекту. Зокрема, зіткнувшись із обмеженнями прямого маніпулювання параметрами рендерера (QgsRuleBasedRenderer), що часто призводить до помилок звернення до атрибутів (setClassAttribute), було застосовано стратегію «зовнішнього управління» шаром.

Такий підхід, згідно з дослідженнями Шермана, полягає у зверненні до об'єктів вищого рівня – дерева шарів (Layer Tree) та API макетів (Layout API). Замість спроб деструктивної модифікації внутрішньої структури рендерера, алгоритм реалізує: використання методу loadNamedStyle для завантаження готових конфігурацій .qml, що забезпечує стабільність візуалізації; застосування методу setSubsetString для прямого SQL-фільтрування даних на рівні провайдера, що дозволяє коректно відображати лише релевантні об'єкти в межах поточного кадру атласу; використання сигналу atlas.featureChanged як тригера для ініціалізації циклу оновлення карти, що є класичним прийомом архітектури PyQGIS для створення складних динамічних систем.

К. Менке при створенні макетів та автоматизації через Atlas наголошує на важливості автоматизації для професійної картографії, щоб уникнути повторюваних ручних дій [11].

Отже, більшість авторів зосереджуються на статичних шаблонах, тоді як метод вкладеної ітерації за

допомогою PyQGIS реалізує автоматичне перехоплення подій зміни сторінки для миттєвого завантаження унікальних файлів стилів (.qml) та жорсткої фільтрації атрибутів. Це дозволяє не просто масштабувати карту, а повністю змінювати її аналітичний зміст (легенду та колірну шкалу) без участі оператора.

**Мета даного дослідження:** мета роботи полягає в оптимізації процесів генерації багатосторінкових тематичних атласів шляхом впровадження алгоритмів автоматизованої стилізації та ітераційної обробки геоданих у середовищі QGIS.

### **Матеріали і методи дослідження**

Досягнення поставленої мети забезпечується впровадженням розробленого авторами підходу до організації вкладених ітерацій, що базується на ієрархічній дворівневій архітектурі управління процесом картографування. На першому рівні первинна ітерація засобами модуля QGIS Atlas забезпечує послідовну обробку просторових об'єктів керуючого шару, здійснюючи автоматичне масштабування, прив'язку геометрії, формування назв та нумерацію сторінок атласу. На другому рівні вторинна ітерація реалізується через Python API (PyQGIS) і відповідає за семантичне управління картографічним відображенням: динамічне завантаження файлів стилів (.qml) методом loadNamedStyle, жорстку фільтрацію просторових об'єктів на рівні провайдера даних методом setSubsetString із застосуванням SQL-запитів, а також оперативне управління шарами-масками в режимі реального часу. Синхронізацію між рівнями ітерації забезпечено через

обробку події `atlas.featureChanged`, яка виступає тригером для ініціалізації циклу оновлення картографічного відображення при переході між об'єктами. Такий підхід реалізує принцип «зовнішнього управління» шаром, при якому модифікація стилістичних параметрів здійснюється через звернення до об'єктів вищого рівня – дерева шарів (Layer Tree) та API макетів (Layout API). Апробацію методу проведено на масиві даних обсягом понад 20 тис. контурів із застосуванням порівняльного аналізу часових витрат між ручним та автоматизованим способами підготовки серії тематичних карт. Оцінку ефективності здійснено на модельному прикладі створення атласу.

### **Результати дослідження та їх обговорення**

Для реалізації методу вкладеної ітерації потрібна підготовка вхідних компонентів: стилів, шарів даних для візуалізації та межі територіального охоплення. Одним із найбільш часозатратних даних є формування набору файлів стилів QGIS (QML). На відміну від статичного оформлення, ці файли містять умовне форматування через логіку розбиття показників на класи чи групи (наприклад, крутизна схилів, якість ґрунтового покриву, типи культур, вид деградації, ступінь придатності до вирощування тощо), налаштовані колірні рампи (Color Ramps), товщину меж та прозорість об'єктів, правила відображення, що дозволяють скрипту миттєво змінювати зовнішній вигляд шару при переході від одного показника до іншого.

Атрибутивно-просторові шари даних, контури полів з геометрією і ко-

ординатною прив'язкою, охоплюють масив інформації, передбачений до візуалізації. Кожне поле атрибутивної таблиці даних просторових шарів може бути використане для візуалізації значень. Передбачається наявність в ньому ключового поля для зв'язку з територіальними межами.

Межі територіального охоплення (Coverage Layers) являє собою управлінський шар атласу (наприклад, межі адміністративних районів, населених пунктів, об'єктів природно-сільськогосподарського районування тощо), який забезпечує просторовий фільтр просторових об'єктів. Саме кожна геометрична одиниця шару стає окремою сторінкою атласу. В той же час просторовий оверлей: шар слугує маскою або межею, яка визначає, які саме об'єкти з основного масиву даних повинні потрапити у поточну компоновку.

Використовуючи вхідні дані скрипт на базі PyQGIS виступає забезпечує їх синергію:

- обирається межа району (оверлейний шар);
- фільтруються за обраною межею контури (шар даних);
- накладається на контури, призначені для візуалізації, відповідна шкала класифікації (файл стилю).

Такий підхід дозволяє перетворити процес, що раніше тривав тижні ручної роботи, на автоматизований цикл, тривалість якого обмежена лише швидкістю рендерингу PDF-файлу. В той же час такий підхід робить систему автоматизованого картографування динамічною та доступною до редагування, в тому числі додавання нових характеристик в просторовий шар та нових стилів вже після налаштування генерування атласу. Для цього достатньо лише

додати новий файл стилю у папку, абсолютно не змінюючи при цьому основний код програми.

Після готовності вхідних даних створюється макет атласу, до якого входять: титульна сторінка, сторінка укладачів, зміст атласу, тематичні карти. Картографічні матеріали оформляються за загальноприйнятими вимогами, зокрема, шаблон сторінки включає назву карти, стрілку напрямку на північ, карту, легенду, масштаб, номер сторінки тощо).

Рівень 3 є найважливішим блоком, так як Python-контролер та запропонований підхід вкладеної ітерації (Processing) забезпечує логіку та автоматизацію створення та динамічного оновлення макету атласу. Реалізація алгоритму базується на подієво-орієнтованій архітектурі PyQGIS, де тригером для виконання коду є системний сигнал зміни активного об'єкта макета. Скрипт перехоплює подію переходу модуля Atlas на новий елемент керуючого шару і автоматично ініціює вкладену процедуру, при якій зчитує атрибутивні дані поточного району, виконує SQL-фільтрацію та динамічно адаптує графічні стилі. На кожному кроці ітерації алгоритм першочергово зчитує атрибутивне ім'я поточного району, що слугує ключо-

вим ідентифікатором для подальшої семантичної обробки. Використовуючи принципи вкладеної логіки, система автоматично звертається до зовнішнього сховища для пошуку відповідного файлу стилів у форматі .qml, назва якого корелює з поточним показником та районом (наприклад, район\_1\_slope.qml). Після ідентифікації потрібного файлу програма ініціює метод loadNamedStyle(), який здійснює миттєве оновлення умовних позначень та кольорових шкал у проекті без необхідності перезавантаження шарів. Паралельно з цим застосовується метод setSubsetString(), який виконує фільтрацію даних через SQL-запити, що дозволяє локалізувати та відобразити виключно ті геометрії (контури), які просторово належать до межі поточного району. Завершальним етапом кожної ітерації є примусове оновлення динамічних елементів макету, таких як текстові заголовки та зміст легенди, після чого сформована сторінка передається на рендеринг, а цикл автоматично переходить до наступного об'єкта. Такий підхід забезпечує повну ідентичність візуальної моделі змісту карти реальному стану вхідних даних для кожної окремої одиниці картографування.

На завершальному етапі функціо-

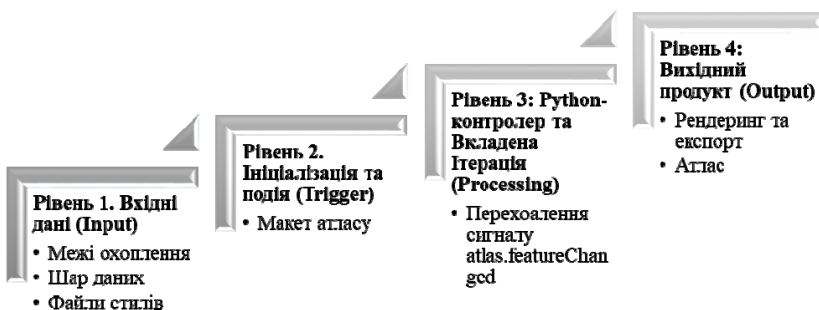


Рис. 1. Концептуальна модель автоматизованого динамічного картографування методом вкладеної ітерації (за допомогою PyQGIS)

нування системи ініціюється автоматизований процес рендерингу та експорту сформованих картографічних матеріалів у цифрові формати високої роздільної здатності, такі як PDF або багатосторінкові растрові зображення. Програмне забезпечення консолідує всі попередньо налаштовані динамічні параметри, забезпечуючи коректну візуалізацію геометрій та їх відповідність метаданим проекту (рис. 1).

Кінцевим результатом цього процесу є цифровий атлас, представлений у вигляді впорядкованої серії карт. Кожна сторінка такого атласу характеризується не лише унікальним просторовим охопленням конкретної адміністративної одиниці, а й персоналізованою візуальною моделлю. Реалізована методика вкладеної ітерації, завдяки якій кожна карта містить індивідуальну шкалу кольорів та умовні позначення, що суворо відповідають специфіці показника (ознаки) для даного району. Такий підхід забезпечує високий рівень інформативності та наочності картографічних матеріалів, мінімізуючи ймовірність суб'єктивних помилок при інтерпретації великих масивів просторових даних (рис. 2).

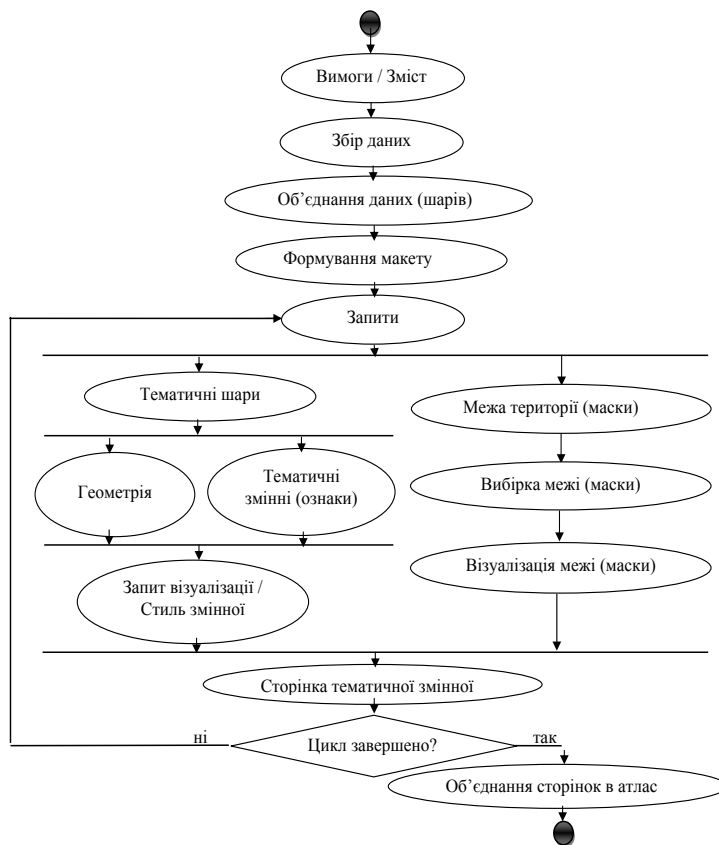
Розроблена функціональна модель відображає процес трансформації вихідних даних у фінальний атлас і складається з трьох ключових етапів:

- підготовчий етап передбачає визначення вимог до змісту, збір та просторове об'єднання тематичних шарів у єдину базу геоданих, що завершується формуванням базового графічного макету в середовищі QGIS;
- етап ітераційної обробки (ядро методу) з дії «Запити» та реалізує паралельну обробку тематичної та територіальної компонент.

Ліва гілка (тематична компонента) алгоритму відповідає за фільтрацію геометрії та семантичних ознак із подальшим динамічним застосуванням стилів (.qml). Вона включає такі дії: формування тематичних шарів, поєднання геометричних примітивів з тематичними змінними, а також запит візуалізації та стиль змінної. Права гілка (територіальна компонента) забезпечує коректне візуальне обмеження території дослідження через механізм маскування меж та її графічного рендерингу для конкретної сторінки. Вона включає такі дії: визначення межі території, вибірка межі (маски) та візуалізація межі;

- на етапі синтезу зациклена структура алгоритму забезпечує автоматичну генерацію сторінок для кожної тематичної змінної (маски), які на фінальному кроці об'єднуються в цілісний атлас. Ключовим елементом управління є вузол прийняття рішень – на схемі логічний ромб для перевірки поставленої умови, – «Цикл завершено?». Перевіряючи чи у базі даних залишаються необроблені геометричні контури, процес за умовним переходом «ні» повертається на новий крок ітерації до блоку «Запити». Якщо умовний перехід фіксує статус «так» (усі територіальні одиниці та показники опрацьовано), алгоритм переходить до фінальної дії – автоматичної консолідації та об'єднання всіх згенерованих сторінок в єдиний цілісний атлас через дію «Об'єднання сторінок в атлас».

Оцінка економії часу при застосуванні автоматизованого динамічного картографування методом вкладеної ітерації порівняно з ручним методом продемонстрована в табл. 1 на прикладі створення атласу для 50 адміністративних одиниць з 5 різними



**Рис. 2. Функціональна модель автоматизованого динамічного картографування методом вкладеної ітерації (за допомогою PyQGIS)**

**1. Порівняльний аналіз часових витрат при ручному та автоматизованому методах картографування**

Етап роботи	Ручний метод (хв/карта)	Автоматизований метод
Підготовка шаблону	60 хв	120 хв (в т.ч. написання коду)
Налаштування фільтрації	5 хв	0 хв (автоматично)
Зміна стилю (.qml)	3 хв	0 хв (автоматично)
Зум та компоунання	2 хв	0 хв (автоматично)
Експорт та нумерація	2 хв	10 хв (на весь проект)
<b>Разом на 250 карт</b>	<b>~3000 хв (50 годин)</b>	<b>~150 хв (2.5 години)</b>

показниками для кожної з них (разом – 250 карт).

Автоматизація дозволяє пришвидшити процес створення серії тема-

тичних карт з однаковою стилістикою у 20 разів. Економія часу складає близько 95% при роботі з великими масивами даних. Застосована мето-

дика дозволяє працювати з великими БД геоданих, не навантажуючи оперативну пам'ять завдяки динамічній SQL-фільтрації.

За більш детальною інформацією про автоматизацію картографування, в тому числі і надання детальних методичних рекомендацій по застосуванню, звертайтеся до авторів статті.

### **Висновки та перспективи**

Впровадження запропонованого підходу вкладеної ітерації у процеси тематичного картографування становить якісний перехід від ручного компонування до повністю автоматизованого інтелектуального середовища генерації геоданих. Застосування зазначеної методики для тематичного картографування дозволяє уникнути людського фактору (помилки при ручній зміні стилів), забезпечує ідентичність оформлення всіх сторінок атласу та вивільняє ресурси фахівця для аналітичної роботи замість технічної підготовки карт.

Результати дослідження доводять, що інтеграція Python-алгоритмів із функціоналом QGIS Atlas дозволяє досягти 20-кратного прискорення підготовки атласів. Запропонований підхід створює надійне технологічне підґрунтя для цифрової трансформації установ, дозволяючи фахівцям фокусуватися на стратегічному аналізі просторових закономірностей, що є критично важливим для сталого управління земельними і іншими природними ресурсами в сучасних умовах.

---

### **Список використаної літератури**

1. Про затвердження Порядку загальнодержавного топографічного і тематичного картографування : Постанова Кабі-

нету Міністрів України від 04 верес. 2013 р. № 661. URL: <https://zakon.rada.gov.ua/laws/show/661-2013-п> (дата звернення: 05.05.2024).

2. QGIS Project. QGIS Documentation: The latest guide to using QGIS for geoprocessing and map creation. URL: <https://qgis.org> (дата звернення: 03.04.2026).
3. Steiniger S., Hunter A. J. The 2012 free and open source GIS software map – A guide to help select the right software for GIS projects. *Computers, Environment and Urban Systems*. 2013. Vol. 39. P. 136–150. DOI: <https://doi.org/10.1016/j.compenvurb-sys.2012.10.003>.
4. Hengl T., Miller B. A., Aldabaa A. A. et al. Digital mapping of soil properties using ensemble machine learning. *Scientific Reports*. 2017. Vol. 7. № 1. P. 1-15. DOI: <https://doi.org/10.1038/s41598-017-18274-6>.
5. Digital Mapping of Soil Properties Using Ensemble Machine Learning Approaches in an Agricultural Lowland Area of Lombardy, Italy / O. D. Adeniyi, A. Brenning, A. Bernini, S. Brenna, M. Maerker. *Land*. 2023. Vol. 12, No. 2. P. 494. DOI: <https://doi.org/10.3390/land12020494>.
6. Artificial Intelligence in Agricultural Mapping: A Review / R. Espinel, G. Herrera, J. L. Rivadeneira Garcia, P. C. Escandón-Panchana. *Agriculture*. 2024. Vol. 14, No. 7. P. 1071. DOI: <https://doi.org/10.3390/agriculture14071071>.
7. Dorosh Y., Dorosh A., Derkul'skiy R., Brat'nova M. Application of GIS in land management on the example of Ukraine. *Acta Scientiarum Polonorum Administratio Locorum*. 2024. Vol. 23 (1). P. 31–41.
8. Balwan Singh. Geographical Information System in Agriculture. *International Journal of Scientific Research in Science and Technology*. 2018. Vol. 4, Issue 2. P. 702–705. URL: [https://academia.edu/37114891/Geographical\\_Information\\_System\\_in\\_Agriculture](https://academia.edu/37114891/Geographical_Information_System_in_Agriculture) (дата звернення: 30.03.2026).

9. Graser A. Learning QGIS. 3rd Edition. Birmingham : Packt Publishing Ltd, 2016. 210 p. URL: [https://books.google.com.ua/books/about/Learning\\_QGIS\\_Third\\_Edition.html?id=rUXicWAAQBAJ](https://books.google.com.ua/books/about/Learning_QGIS_Third_Edition.html?id=rUXicWAAQBAJ) (дата звернення: 03.04.2026).
10. Sherman G. The PyQGIS Programmer's Guide: Extending QGIS 3 with Python 3. Locate Press, 2018. 252 p. URL: <https://locatepress.com/book/ppg3> (дата звернення: 03.04.2026).
11. Menke K. Mastering QGIS. 2nd Edition. Birmingham : Packt Publishing, 2019. 404 p.

### References

1. Cabinet of Ministers of Ukraine. (2013, September 4). Pro zatverdzhennia Poriadku zahalnodержавного topohrafichnoho i tematychnoho kartohrafuvannia [On approval of the Procedure for national topographic and thematic cartography] (Resolution No. 661). Available at: <https://zakon.rada.gov.ua/laws/show/661-2013-n>
2. QGIS Project. (n.d.). QGIS documentation: The latest guide to using QGIS for geoprocessing and map creation. Available at: <https://qgis.org>
3. Steiniger, S., & Hunter, A. J. (2013). The 2012 free and open source GIS software map – A guide to help select the right software for GIS projects. *Computers, Environment and Urban Systems*, 39, 136–150. DOI: <https://doi.org/10.1016/j.compenurbysys.2012.10.003>
4. Hengl, T., Miller, B. A., Aldabaa, A. A., et al. (2017). Digital mapping of soil properties using ensemble machine learning. *Scientific Reports*, 7(1), 1–15. DOI: <https://doi.org/10.1038/s41598-017-18274-6>
5. Adeniyi, O. D., Brenning, A., Bernini, A., Brenna, S., & Maerker, M. (2023). Digital mapping of soil properties using ensemble machine learning approaches in an agricultural lowland area of Lombardy, Italy. *Land*, 12(2), 494. DOI: <https://doi.org/10.3390/land12020494>
6. Espinel, R., Herrera, G., Rivadeneira Garcia, J. L., & Escandón-Panchana, P. C. (2024). Artificial intelligence in agricultural mapping: A review. *Agriculture*, 14(7), 1071. DOI: <https://doi.org/10.3390/agriculture14071071>
7. Dorosh, Y., Dorosh, A., Derkul'skiy, R., & Bratinova, M. (2024). Application of GIS in land management on the example of Ukraine. *Acta Scientiarum Polonorum Administratio Locorum*, 23(1), 31–41.
8. Singh, B. (2018). Geographical information system in agriculture. *International Journal of Scientific Research in Science and Technology*, 4(2), 702–705. Available at: [https://academia.edu/37114891/Geographical\\_Information\\_System\\_in\\_Agriculture](https://academia.edu/37114891/Geographical_Information_System_in_Agriculture)
9. Graser, A. (2016). Learning QGIS (3rd ed.). Birmingham: Packt Publishing Ltd, 210. Available at: [https://books.google.com.ua/books/about/Learning\\_QGIS\\_Third\\_Edition.html?id=rUXicWAAQBAJ](https://books.google.com.ua/books/about/Learning_QGIS_Third_Edition.html?id=rUXicWAAQBAJ)
10. Sherman, G. (2018). The PyQGIS programmer's guide: Extending QGIS 3 with Python 3. Locate Press, 252. Available at: <https://locatepress.com/book/ppg3>
11. Menke, K. (2019). Mastering QGIS (2nd ed.). Birmingham: Packt Publishing, 404.

**Dorosh Y., Zaiachkivska B., Kharytonenko R., Melnyk D.**

### **APPLICATION OF THE NESTED ITERATION METHOD FOR AUTOMATED MAPPING**

*LAND MANAGEMENT, CADASTRE AND LAND MONITORING 2'26: 4-14*

<http://dx.doi.org/10.31548/zemleustriy2026.02.01>

**Abstract.** The article considers a method for automating the creation of a series of cartographic materials based on the integration of QGIS Atlas tools and the Python programming language. The

*study aims to optimize the preparation process of multi-page thematic atlases for large spatial datasets. An original approach to applying the nested iteration technique has been introduced, which allows for dynamic changes not only in the spatial extent but also in the semantic styling of the display for each individual object. A key element is the developed algorithm based on the processing of the featureChanged event, which ensures automatic synchronization of district attributes with the corresponding .qml style files. The model was tested on a dataset exceeding 20,000 contours.*

*The testing results confirmed the high efficiency of the method. The speed of map series generation increased approximately 20-fold compared to traditional methods, while completely eliminating subjective visualization errors. The proposed approach represents a universal methodological platform for the automated transformation of complex spatial datasets into high-precision graphical models. This solution provides the capability for rapid presentation of multifactor research results in any field requiring a combination of high-intensity processing of large datasets with adherence to individual visual interpretation parameters for each unique system element.*

**Keywords:** *automated mapping, QGIS Atlas, PyQGIS, agrochemical monitoring, GIS automation, nested iteration method, dynamic styling, thematic atlases, Python API, \*.qml style files, layout automation, mapping algorithmization.*

---