

МЕТОД АДАПТИВНОГО УПРАВЛІННЯ ЖИТТЄВИМ ЦИКЛОМ ІНТЕЛЕКТУАЛЬНИХ АГЕНТІВ У МУЛЬТИДОМЕННИХ ITSM-СИСТЕМАХ

Вінницький національний технічний університет;

Анотація

У роботі досліджено проблему організації життєвого циклу інтелектуальних агентів, що функціонують у складі ITSM-систем з мультидоменим розгортанням. Класичні моделі життєвого циклу програмного забезпечення не враховують особливостей еволюції AI-компонент, специфіку доменно-залежних конфігурацій та потребу безперервного оцінювання якості агентів після введення в експлуатацію. Запропоновано метод адаптивного управління життєвим циклом, що поєднує практики DevOps та MLOps з підходом оцінювально-керованого розроблення і механізмами ізоляції доменних артефактів. Сформульовано основні фази запропонованого методу та визначено функціональні компоненти програмного середовища, яке підтримує його реалізацію.

Ключові слова: життєвий цикл програмного забезпечення; AIOps; ITSM; мультидомenna архітектура; інтелектуальні агенти; MLOps; DevOps.

Abstract

The paper investigates the problem of organising the lifecycle of intelligent agents operating within ITSM systems deployed across multiple tenant domains. Conventional software lifecycle models do not capture the evolutionary nature of AI components, domain-specific configuration requirements, or the need for continuous quality evaluation after release. A method of adaptive lifecycle management is proposed, combining DevOps and MLOps practices with the evaluation-driven development principle and tenant-domain artefact isolation mechanisms. The principal phases of the method are formulated, and the functional modules of a software environment supporting its implementation are defined.

Keywords: software lifecycle; AIOps; ITSM; multi-tenant architecture; intelligent agents; MLOps; DevOps.

Вступ

Сучасні платформи управління IT-послугами (ITSM), що використовують технології штучного інтелекту, поступово витісняють традиційні підходи до автоматизації обробки інцидентів, обслуговування запитів та супроводу конфігураційних одиниць. Інтеграція великих мовних моделей та автономних агентів у процеси, регламентовані бібліотекою ITIL, дозволяє зменшити час реагування на інциденти та знизити навантаження на операторів першої лінії технічної підтримки.

Водночас комерційне впровадження подібних платформ зазвичай реалізується за моделлю SaaS з обслуговуванням множини клієнтських організацій (доменів) на спільній інфраструктурі. Такий формат розгортання породжує клас задач, які слабо охоплюються традиційними моделями життєвого циклу програмного забезпечення, оскільки кожен клієнт має власний набір системних промтів, інструментів-конекторів та політик доступу, що повинні версіюватися окремо від ядра платформи.

Актуальність

Актуальність дослідження зумовлена стрімким розвитком AI-driven ITSM-платформ, що обслуговують множини клієнтських організацій на спільній SaaS-інфраструктурі. Поведінка інтелектуальних агентів у складі таких платформ є недетермінованою та схильною до деградації внаслідок дрейфу базових моделей, оновлення промтів і змін у характері звернень користувачів. Водночас кожен домен потребує власних налаштувань агентів, інтеграцій з інструментами та політик безпеки, що мають окремий цикл розвитку від ядра платформи.

У класичних SDLC-моделях ці аспекти не отримують самостійного представлення, що призводить до накопичення технічного боргу та зниження керованості випусків. Недостатньо опрацьована послідовність етапів життєвого циклу AI-компонент може призвести до некерованої деградації якості сервісу та ускладнень із дотриманням SLA. Тому актуальним є розроблення моделей і процесів життєвого циклу, спеціально адаптованих для AI-driven ITSM-систем з мультидоменим обслуговуванням.

Метод адаптивного управління життєвим циклом інтелектуальних агентів

Сучасні AI-driven ITSM-системи інтегрують технології машинного навчання та великих мовних моделей у процеси, регламентовані бібліотекою ITIL, що дозволяє автоматизувати класифікацію та обробку звернень користувачів [1]. У зв'язку зі зростанням обсягу інцидентів та підвищенням вимог до якості сервісного обслуговування, дослідники наголошують на потребі інтегрованих рішень, що поєднують класифікацію звернень, оптимізацію розподілу ресурсів та пояснюваність ухвалених рішень у єдиній архітектурній рамці [2].

Особливості мультидоменного розгортання інтелектуальних агентів вимагають додаткових механізмів ізоляції контекстів, версіонування доменно-залежних артефактів та запобігання витоку даних між клієнтами на рівні модельної пам'яті [3]. У промислових конвеєрах підготовки інтелектуальних компонент, зокрема MLOps, акцент робиться на циклах обробки даних, навчання моделі, валідації та супроводу [4]. Розширення цих практик для агентських систем, таких як LLMOps або AgentOps, дозволяє покрити роботу з промтами, інструментами та пам'яттю агента, проте мультидоменна специфіка залишається відкритою задачею.

Для подолання зазначених обмежень пропонується метод адаптивного управління життєвим циклом агентів, у якому оцінювальні процедури вбудовуються безпосередньо у процес розроблення та супроводу. Підхід поєднує офлайн-оцінювання на репрезентативних наборах сценаріїв з онлайн-моніторингом якості рішень агента під час експлуатації, що відповідає принципам evaluation-driven development [5]. Ключовою архітектурною особливістю методу є виділення доменно-залежного шару артефактів (системних промтів, інструментів-конекторів, вектор-індексів знань та політик доступу), який має власний субжиттєвий цикл, синхронізований із основним.

Метод визначає такі основні етапи життєвого циклу інтелектуального агента:

- 1) формалізація бізнес-сценарію та критеріїв оцінювання якості агента;
- 2) побудова базового шаблону агента: системний промт, набір інструментів, джерела знань;
- 3) доменна параметризація шаблону для конкретного клієнта;
- 4) офлайн-оцінювання з подальшим канарковим випуском на частку трафіку;
- 5) онлайн-моніторинг якості рішень, витрат токенів та поведінки інструментів;
- 6) ітеративне доопрацювання промтів і конфігурацій за результатами оцінювання;
- 7) міграція на нову версію базової моделі або виведення агента з експлуатації.

Для зіставлення запропонованого методу з відомими підходами здійснено порівняльний аналіз ключових моделей супроводу програмних рішень з компонентами штучного інтелекту, результати якого наведено у таблиці 1.

Таблиця 1. Порівняння моделей життєвого циклу для AI-компонент у ITSM-системах

Модель	Сфера застосування	Сильні сторони	Обмеження для ITSM-агентів
Класичний DevOps	Безперервна інтеграція коду	Зрілі інструменти, швидкі випуски	Не охоплює дрейф моделі та якість відповідей
MLOps	ML-моделі прогнозування	Відтворюваність, автоматичне дотренування	Слабко покриває агентську поведінку та інструменти
LLMOps / AgentOps	Промпт-інженерія, оцінювання LLM	Оцінювальні набори, спостережуваність	Не враховує мультидоменну ізоляцію артефактів
Запропонований метод	AI-driven ITSM з мультидоменним розгортанням	Доменний шар артефактів, синхронізовані субжиттєві цикли	Потребує організаційної зрілості замовника

На основі сформульованих етапів розроблено архітектуру програмного середовища, яке реалізує запропонований метод. Основні функціональні модулі такого середовища наведено у таблиці 2.

Таблиця 2. Функціональні модулі системи підтримки життєвого циклу AI-агентів

Модуль	Функції
Реєстр шаблонів агентів	Зберігання версіонованих базових конфігурацій агентів
Менеджер доменних артефактів	Версіонування промтів, інструментів і знань конкретного клієнта
Конвеєр офлайн-оцінювання	Виконання тест-наборів та обчислення метрик якості агента
Оркестратор канаркового випуску	Поетапне розкочування нової версії на частку трафіку домену
Сервіс онлайн-спостережуваності	Збір телеметрії, оцінок користувачів і логів інструментів
Детектор дрейфу та регресій	Виявлення відхилень і запуск процедур відкату або оновлення

Реалізація запропонованої архітектури дає змогу скоротити час від виявлення проблеми у поведінці агента до її усунення, забезпечити незалежне версіонування артефактів кожного клієнтського домену та підтримати повторюваність релізів у мультидоменному середовищі.

Висновки

У роботі проаналізовано особливості життєвого циклу інтелектуальних агентів, що функціонують у складі AI-driven ITSM-систем з мультидоменним розгортанням. Показано, що наявні моделі життєвого циклу програмного забезпечення лише частково задовольняють вимоги до супроводу таких систем, оскільки не охоплюють специфіки доменно-залежних артефактів, недетермінованої поведінки агентів і необхідності безперервного оцінювання якості.

Запропоновано метод адаптивного управління життєвим циклом, що інтегрує практики DevOps, MLOps та оцінювально-керованого розроблення з виділеним доменно-залежним шаром артефактів. Сформульовано основні етапи методу та визначено функціональні модулі програмного середовища, що підтримує його реалізацію. На відміну від еталонних агентських AIOps-фреймворків, запропонований метод явно враховує мультидоменну специфіку та доменно-залежне версіонування артефактів. Отримані результати можуть бути використані під час проєктування комерційних AI-driven ITSM-платформ та слугувати основою для подальших досліджень моделей і процесів життєвого циклу інтелектуальних інформаційних систем[6].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hu J., Yan Y., Zhao Y., Wang Z., Lin Y. та ін. «A Survey of AIOps in the Era of Large Language Models», DOI: 10.1145/3746635, 2025. URL: <https://arxiv.org/abs/2507.12472>
2. Peralta A., Olivas J. A., Romero F. P., Navarro-Illana P. «Intelligent Incident Management Leveraging Artificial Intelligence, Knowledge Engineering, and Mathematical Models in Enterprise Operations», DOI: 10.3390/math13071055, 2025. URL: <https://www.mdpi.com/2227-7390/13/7/1055>
3. Roshan A. та ін. «Multi-Tenant Isolation Challenges in Enterprise LLM Agent Platforms», 2025. URL: https://www.researchgate.net/publication/399564099_Multi-Tenant_Isolation_Challenges_in_Enterprise_LLM_Agent_Platforms
4. Steidl M., Felderer M., Ramler R. «The Pipeline for the Continuous Development of Artificial Intelligence Models: Current State of Research and Practice», DOI: 10.48550/arXiv.2301.09001, 2023. URL: <https://arxiv.org/abs/2301.09001>

5. Xia B., Lu Q., Zhu L., Xing Z., Zhao D., Zhang H. «Evaluation-Driven Development of LLM Agents: A Process Model and Reference Architecture», DOI: 10.48550/arXiv.2411.13768, 2024. URL: <https://arxiv.org/abs/2411.13768>

6. Zota R. D., Bărbulescu C., Constantinescu R. «A Practical Approach to Defining a Framework for Developing an Agentic AIOps System», DOI: 10.3390/electronics14091775, 2025. URL: <https://www.mdpi.com/2079-9292/14/9/1775>

Педосенко Денис Володимирович - студент групи ІПІ-25м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: dpedosenko@gmail.com

Науковий керівник: Ракитянська Ганна Борисівна - канд. техн. наук, доцент, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: rakit@vntu.edu.ua

Pedosenko Denys Volodymyrovych - student of group ІPI-25m, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: dpedosenko@gmail.com

Academic supervisor: Rakityanska Hanna Borysivna - Cand. Sc. (Eng.), Associate Professor, Associate Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: rakit@vntu.edu.ua