

Low Code – не тільки для прототипів: аргументи «за» та «проти»

—

kpmg.ua

01

**Найпоширеніші
аргументи проти
Low Code enterprise
розробки**

«Low Code – це тільки для прототипів»

Аргументи «За»



Обмежена
гнучкість



Проблеми з
інтеграцією



Складність для
ринку України



Обмежене
масштабування



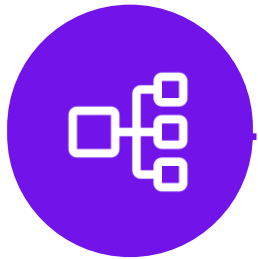
Залежність від
постачальника

Обмежена гнучкість



Складні та спеціалізовані завдання

Low Code платформи можуть не надавати потрібну гнучкість для вирішення дуже спеціалізованих або складних завдань, які вимагають налаштувань на низькому рівні



Продуктивність

Системи, побудовані на Low Code платформах, можуть мати проблеми з продуктивністю через надлишковий код або неоптимізовані запити до бази даних



Технічні борги

У довгостроковій перспективі, рішення, створені на Low Code платформах, можуть накопичувати технічні борги через обмежені можливості оптимізації та рефакторингу

Проблеми з інтеграцією

Конектори – це єдиний варіант у Low Code інструментах для інтеграції з third-party?

3 види конекторів з third-party, які надають Low Code вендори

Пряма black-box інтеграція

- Використання передналаштованих інтеграцій із певними сервісами, що надає можливість швидше під'єднатися до певного переліку застосунків.
- Це дуже обмежує потенційний функціонал цільового застосунку, але робить систему більш простою та передбачуваною.

Класичні варіанти інтеграції

- Використання REST / SOAP / GraphQL для запитів до зовнішніх сервісів / серверів
- Надання користувачу Тьюринг-повного редактора поведінки back-end логіки

Інтеграція через єдиного провайдера (рідше за все)

- Актуально для front-end only інструментів.
- Використовують один інструмент, через який реалізується логіка back-end, але цей інструмент ніяк не пов'язаний з поточним.

Складність сприйняття на ринку України



За що звикли платити в розробці продуктів web/mobile - app:

- Час розробників
- Third-party сервіси, які використовуються для функціоналу застосунку
- PM/QA витрати

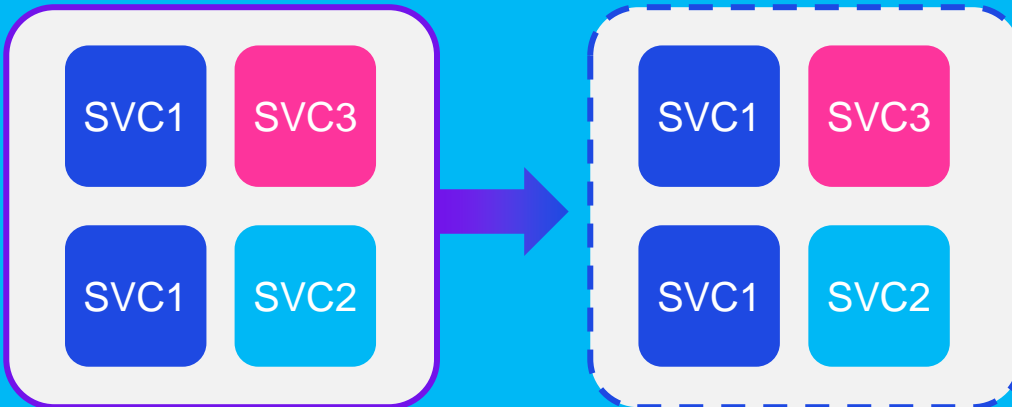


За що не звикли платити в розробці продуктів web/mobile - app:

- Ліцензії інструменту для розробки
- Функції, які неможливо не включити в пакет ліцензії
- User-based зміну ціни за ліцензію

Обмежене масштабування

Відсутність явної підтримки мікросервісної архітектури



- Відсутність брокерів повідомлень
- Неявна асинхронність процесів
- Відсутність керування моделлю масштабування

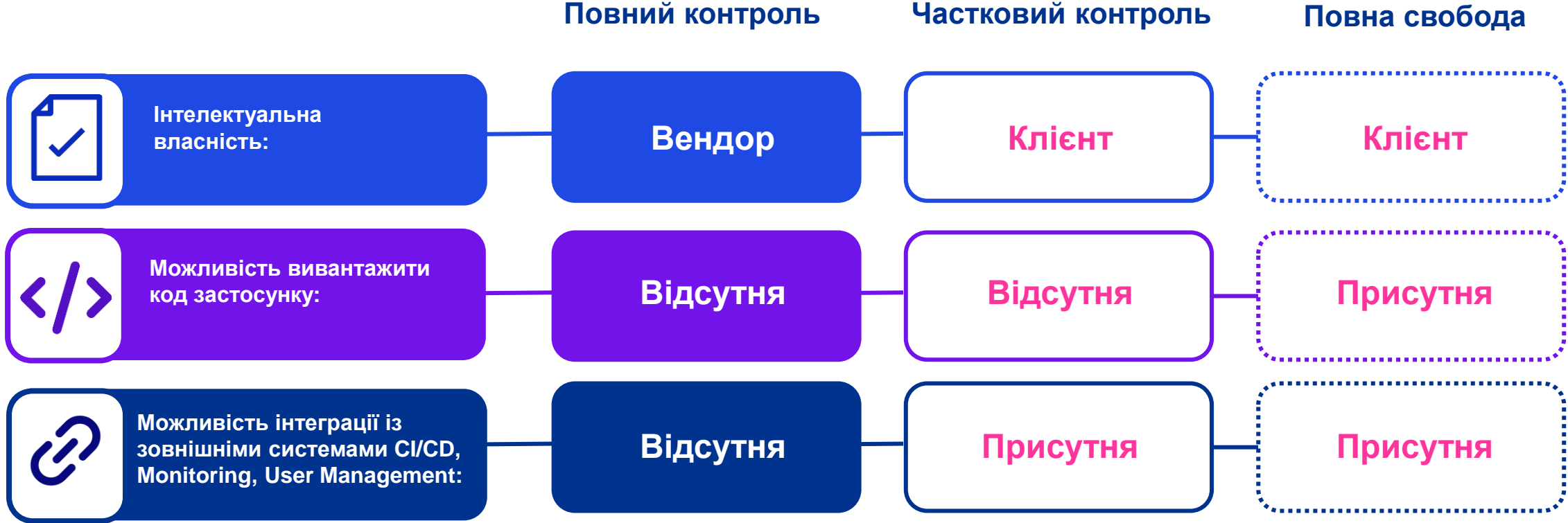
Неможливість або надвисока складність документації

- Неможливість автоматизувати документування через закритість системи

Відсутність / відходження від сталих стандартів розробки

- Відсутність FaaS для back-end
- Відсутність стандартних методів авторизації та стандартів написання API
- Не стандартні назви компонентів (Наприклад: Не асинхронні процеси – а Actions)

Залежність від постачальника



02

Принципи вибору інструментів

Критерії вибору інструмента

Розподілений front/back-end

Це дасть вам перевагу, оскільки ваш клієнт не буде платити за те, що йому не потрібно, що дає гнучкість у вашій комерційній пропозиції.

Хостинг рішення за стандартом на публічних хмарах

Хостинг на публічних хмарах надасть інструменту можливість більш нативно інтегруватися із хмарою та підвищить ймовірність використання провайдером коректних технік auto-scaling.

Наявність імплементованих стандартів класичної сфери розробки

Використання вендором стандартів ринку з боку авторизації, менеджменту back/front-end стандартів надасть можливість клієнту більше зрозуміти продукт та розробникам легше знаходити інформацію.

Вбудовані та повні можливості розробки власних компонентів та кастомного коду

Дуже часто вендори не дають повноцінно розробляти компоненти, або роблять цей процес надто важким. Вибір більш відкритої платформи надасть гнучкості.

Максимальна сепарованість від вендора

Використання Low Code не повинно означати некомфортну прив'язку до вендора.

Права на власність продукту, можливість забрати код з платформи – це ті чинники, на які варто звернути увагу.

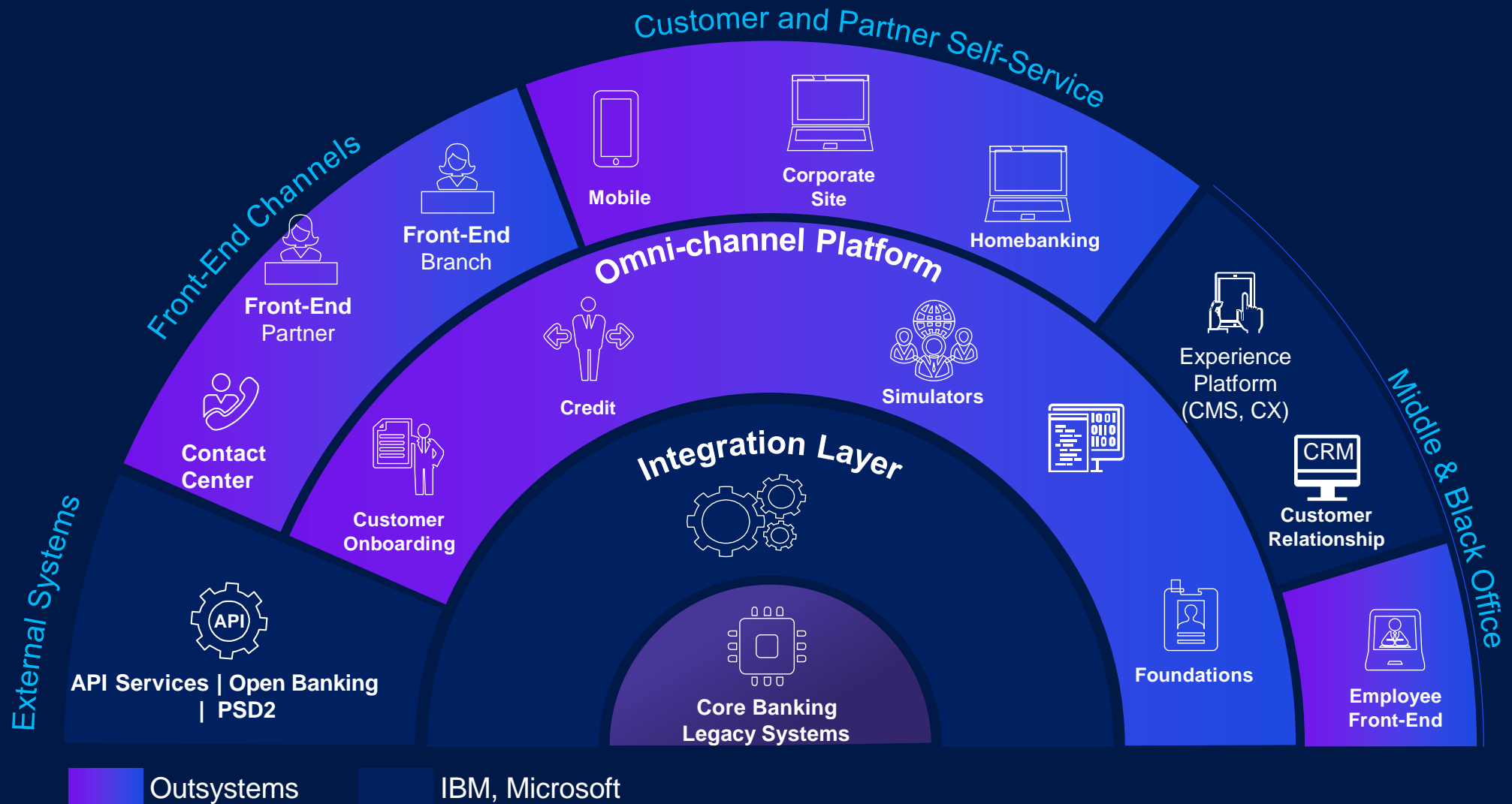
Критерії вибору інструмента



03

**Low Code для
Enterprise рішень:
Кейс**

Розробка банківської інфраструктури



Впроваджений функціонал



Оцінка поточної операційної моделі IT,
визначення можливих сценаріїв отримання даних, визначення мапи шляхів цифрової трансформації та визначення вимоги до IT-операційної моделі для підтримки моделі 'run the bank' та 'change the bank'.



Реалізація нової IT-архітектури в хмарній інфраструктурі, включаючи новий інтеграційний шар (Microsoft APIM)



Впровадження уніфікованого рішення за допомогою Low Code інструменту OutSystems для взаємодії з бізнес-процесами (наприклад, кредитування, онбординг і KYC, кредитні симулятори, CRM і рейтинг) і нових цифрових каналів (веб + мобільний банкінг, PSD2 і відкритий банкінг) для роздрібних і корпоративних клієнтів.

Спікер



Андрій Войналович

Менеджер, послуги у сфері впровадження та підтримки ERP систем та інформаційних технологій

KPMG в Україні

E: avoinalovych@kpmg.ua

T: +380 506 47 1408





Аудиторським клієнтам KPMG та їх афілійованим або пов'язаним особам може бути заборонено надання деяких або всіх описаних у цьому документі послуг.



kpmg.ua

Інформація, що подана у цій публікації, носить загальний характер і не висвітлює стан справ будь-якого окремого підприємства або фізичної особи. Незважаючи на те, що ми намагаємося подавати точну і своєчасну інформацію, ми не гарантуємо, що ця інформація є правильною на дату її отримання або буде достовірною у майбутньому. Ніхто не повинен діяти і покладатися на таку інформацію без відповідної професійної консультації, наданою після детального вивчення стану справ.

Назва KPMG та логотип KPMG є зареєстрованими торговими марками або товарними марками, що використовуються за ліцензією учасниками глобальної організації незалежних фірм KPMG.

© 2023 ТОВ «КПМГ-Україна», компанія, зареєстрована згідно із законодавством України, член глобальної організації незалежних фірм KPMG, що входять до KPMG International Limited, приватної англійської компанії з відповідальністю, обмеженою гарантіями своїх учасників. Усі права застережені.

Статус документа: Публічно