

DOI: <https://doi.org/10.32782/2524-0072/2024-65-78>

УДК 004.415.5, 65.012

УПРАВЛІННЯ ЯКІСТЮ: МЕТОДОЛОГІЯ ВІДБОРУ ІНСТРУМЕНТАРІЮ ДЛЯ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЧЕРЕЗ PROOF OF CONCEPT TA BENCHMARKING

QUALITY MANAGEMENT: A METHODOLOGY FOR SELECTING SOFTWARE TESTING TOOLS USING PROOF OF CONCEPT & BENCHMARKING

Ільницький Ігор Олегович

магістр з менеджменту організацій,

власник бізнесу у сфері ІТ,

співзасновник і голова громадської організації,

Івано-Франківський національний технічний університет нафти і газу

ORCID: <https://orcid.org/0009-0000-0464-8275>

Ilnytskyi Ihor

Ivano-Frankivsk National Technical University of Oil and Gas

Бізнес, менеджмент та спеціалісти у повсякденному житті компанії мають постійно вирішувати різноманітні задачі для досягнення цілей організації і приймати відповідні управлінські рішення щодо вибору найкращого способу. Є різні методи та манери вирішення питань бізнесу, та ефективнішим буде використання наукового підходу у поєднанні з професіоналізмом та практичним досвідом. У множині аспектів ведення бізнесу пріоритетним є, окрім виготовлення продукції (надання послуг), забезпечення якості. Забезпечення якості – це невід’ємна складова пропозиції компанії, у якій зацікавлені замовники та кінцеві споживачі. Якість включає набір різних компонентів, які потрібно виконати. Для вибору технології виконання використовують різноманітний інструментарій, що допомагає прийняти відповідні рішення. У даній статті наведемо реальний, проте дещо абстрагований приклад використання методу доведення концепції (Proof of Concept, PoC), що включає методику порівняння альтернатив (Benchmarking), щоб вибрати найкращий тестовий фреймворк (test framework), який стане основою АQA, оскільки, як відомо, автоматизація – це значно ефективніше рішення реалізації QA діяльності. Наявність автоматичних процесів позитивно впливає на інвестиційну привабливість і ринкову ціну (капіталізацію) підприємства. Прив’язка практичного прикладу до відповідних методологій та термінології дає краще розуміння суті способу. PoC разом з Benchmarking забезпечують універсальний науковий підхід до вирішення не тільки наведених завдань, їх також можна використовувати для інших задач. Результат дослідження ідеї оформляється відповідним комплексним документом, який розглядається та обговорюється на нараді (meeting), організованій для прийняття рішення щодо вибору, а також може стати взірцем для виконавців в подальшому і бути продемонстрованим для обґрунтування менеджменту та іншим зацікавленим сторонам. Застосування підходу має вищий поріг впровадження при першому використанні, проте, в кінцевому результаті, збереже час і кошти, забезпечить темп та покращить експертизу (спеціалізацію) компанії. Одним з альтернативних варіантів прийняття рішень є бізнес-план, проте має свої цілі і це тема окремої наукової праці.

Ключові слова: прийняття управлінських рішень, доведення концепції, порівняльний аналіз, порівняльне оцінювання, управління якістю, автоматизація тестування, інструменти автоматизації, автоматизація забезпечення якості, науковий підхід, управління, менеджмент, господарювання, економіка, інформаційні технології, ІТ.

Business, management and specialists inside the company constantly face various challenges in order to achieve the goals of the organization and have to make appropriate decisions regarding the choice of the best solution. There are different methods and manners of solving business issues, and it will be more effective to use a scientific approach combined with professionalism and practical experience. Running a business involves many aspects, among which, beyond manufacturing products and providing services, quality assurance is a priority. Quality assurance is an integral part of the company's offer, in which customers and end consumers are interested. Quality includes a set of different components that need to be fulfilled. Various tools might be helpful to select an option of the implementation technology, which assist to make the appropriate decisions. In the article, we will give a practical however somewhat abstracted example of applied the Proof of Concept (PoC) method, which comprises

the Benchmarking methodology, to choose the best test framework that will become the basis of AQA, since, as is known, automation is a much more efficient solution of implementation of QA activities. The presence of automatic processes has a positive effect on the investment attractiveness and market price (capitalization) of the enterprise. Linking a practical example to relevant methodologies and terminology gives a better understanding of the essence of the method. PoC together with Benchmarking provide a universal scientific approach to solve not only the above tasks, they can also be used for other issues. The result of an idea's research is formalized in an appropriate comprehensive document, which is reviewed and discussed at a meeting organized to make a decision on the choice, can become a template for the executors in the future and be demonstrated to justify the management and other concerned parties. The application of the approach has a higher threshold of implementation at the first use, but, in the end, it will save time and money, provide the pace and improve the expertise (specialization) of the company. One of the alternative options for decision-making is a business plan, but it has its own goals and is the subject of a separate scientific work.

Key words: making management decisions, Proof of Concept, PoC, Benchmarking, quality management, QM, TQM, test automation, automation tools, automation quality assurance, AQA, scientific approach, management, economics, information technologies, IT.

Постановка проблеми. Прийняття вірних обґрунтованих управлінських рішень є важливим чинником успіху господарювання. Ключовим осердям таких рішень має бути науковий підхід, який можна описати такими складовими процедурами:

- ідентифікація та виокремлення проблеми (задачі);
- збір та аналіз даних для вивчення питання;
- формування альтернативних рішень;
- порівняльна оцінка альтернатив (до прикладу Benchmarking);
- вибір найкращого рішення на базі порівняння описів та величин;
- імплементація рішення;
- постійний моніторинг та оцінка результатів.

Все вищенаведене разом є описом варіанту реалізації наукового підходу в управлінні під назвою доведення концепції (Proof of Concept, PoC). Для кращого розуміння методології, для власників бізнесу, менеджерів різного рівня та спеціалістів, задіяних у реалізації рішень, допоміжним є вивчення прикладів вирішення задачі вищенаведеним чином. Тому дана наукова праця покликана надати зразок, певного рівня узагальнення, реалізації підходу на практиці у сфері інформаційних технологій, а саме AQA, що є частиною TQM.

Аналіз останніх досліджень і публікацій. Різноманітні джерела пропонують багато теоретичної та практичної інформації по наведеній тематиці. Дана стаття пропонує приклад власного досвіду реалізації підходу PoC та Benchmarking у прийнятті управлінських рішень по вирішенню задач автоматизації тестування.

Виділення невирішених раніше частин загальної проблеми. Прийняття управ-

лінських рішень – це відповідальний процес вибору опцій, що є складовими досягнення цілей. Відбір може бути проведено реалізацією першої знайденої ідеї, випадковим чином серед набору варіантів, виходячи з відчуттів інтуїції, на основі практичного досвіду, через застосування наукового підходу, проте зазвичай у свої діяльності менеджери поєднують наведені методи. Також слід зазначити, що обрання робиться індивідуально, групою людей чи ієрархічно. Найкращим буде поєднання професійних навичок та інтуїції, практичного досвіду і наукових способів індивідуумами чи групою людей. У даній статті буде наведено реальний та дієвий практичний приклад, певного рівня абстракції, що стосується реалізації наукового підходу при обранні інструменту автоматизації тестування, що включатиме ефективні методики Proof of Concept та Benchmarking.

Мета статті. Метою дослідження є забезпечити практичним прикладом обрання засобу автоматизації тестування за допомогою поєднання методів Proof of Concept та Benchmarking. Дана інформація, безпосереднє застосування методології та результати примінення на практиці будуть корисними для всіх учасників процесу як всередині компанії так і за її межами.

Виклад основного матеріалу дослідження. Забезпечення якості є визначальним та безвиключним складником продукту у IT і будь-яких інших компаніях матеріальної та нематеріальної сфери. Для перевірки виконання функціональних вимог і тих, що не пов'язуються з функціоналом (такі як дизайн, продуктивність, безпека та інші), буде ефективнішим використовувати різноманітні інструменти автоматизації, що допомагають прискорити отримання результатів і забезпечують багаторазове виконання тестів про-

тягом дня, а також покращують інвестиційну привабливість і збільшують ринкову ціну (капіталізацію) компанії. Для цього, на початковому етапі або приймається рішення про використання власних, вже існуючих рішень, або ж обирають підходяще ПЗ серед доступних пропозицій. Третім, альтернативним варіантом може бути реалізація необхідного засобу всередині компанії-виконавця замовлення, що є довготривалим, затратним і не завжди відповідає суті бізнесу. Є різні підходи як обрати інструментарій для автоматичного тестування, та найпрогресивнішим буде використання універсального методу доказу концепції (Proof of Concept, PoC). Основою вищезазначеної методології буде табличне порівняння 2 чи більше опцій, тобто бенчмаркінг (Benchmarking), з наступним обранням найкращої. Для кращого розуміння наведу лаконічні визначення ключових термінів:

- *доказ концепції* – обґрунтування життєздатності рішення;
- *бенчмаркінг* – співставлення, в тому числі з еталоном, для здійснення порівняльної оцінки, вибору, прийняття ефективних управлінських рішень.

Для цілей даної статті можна виділити наступні складові розділи PoC:

- постановка проблеми – опис необхідних активностей, важливість для бізнесу;
- припущення – опис ідеї;
- обмеження:
 - вимоги – опис/список вимог до продукту;
 - бізнес-обмеження – домовлені правила використання середовищ, ліцензійні ліміти тощо;
 - технічні обмеження – особливості спільного використання середовищ, необхідність очищення тестових середовищ, вимоги до даних, години доступності інфраструктури, обов'язкова нотифікація зацікавлених сторін стосовно переконфігурування середовищ, відключення незадіяних, інтеграцій і початку та завершення тестування та інше;
- порівняння опцій – табличне порівняння оцінок інструментів, тобто безпосередньо бенчмаркінг;
- рекомендації – порівняльне обрання найкращого варіанту;
- наслідки – ризики, подальші необхідні додаткові зусилля в результаті обрання, прогнози;
- список дій – що потрібно зробити для втілення ідеї, відповідні відповідальні особи;
- файли та документація – допоміжна вхідна та вихідна інформація.

Практичний абстрагований приклад PoC із зінтегрованим Benchmarking матиме наступний вигляд.

Постановка проблеми. Кожен продукт вимагає бути перевіреним на відповідність визначеним та потенційним вимогам перед переведенням у роботу в реальних умовах. Для наведених цілей необхідно імплементувати тестовий фреймворк для виконання AQA. Автоматизація допоможе швидко отримувати результати тестування та виконувати перевірки багаторазово протягом дня. Наявність автоматизації позитивно впливає на інвестиційну привабливість та збільшує ринкову ціну (капіталізацію) бізнесу.

Припущення. Для виконання завдання необхідно розглянути кілька провідних технологій та порівняти між собою підходящі варіанти з метою обрання найкращого. Є багато популярних інструментів для автоматизації та найбільш підходящими для нашого проекту будуть: [перелік]. Для порівняння, відбору і подальшого використання оберемо [два інструменти з назвами мов програмування і причиною відбору]. Запуск тестів відбуватиметься на щоденній основі по графіку, можливо за потреби кількаразово, у віддаленому репозиторії для спільного виконання завдань і загального доступу зацікавлених осіб до результатів для якнайшвидшого прийняття рішень щодо автоматизації, виконання (оновлення, виправлення) та випуску продукту, управління та бізнесу. Перелік програм (функціоналу) для тестування наводимо у таблиці: [Табличний перелік]. Дизайн виконання тестів наведено на рисунку: [Рисунок]. Не включено або відкладено наступні програми (функціонал): [Список]. Опис типу тестування, тривалості одного пробігу тестів та інших гіпотетичних параметрів.

Обмеження.

Вимоги. Продукт має відповідати наступним критеріям прийомки ПЗ, що наведені у доданій таблиці. [Таблиця чи Посилання].

Обмеження бізнесу: [Перелік середовищ і для чого мають використовуватись, з ким погоджувати активності перед запуском на певних середовищах, ліцензування і можливості по ліцензіях, ключові персони для погоджень, що неможливо виконати з точки зору бізнесу тощо].

Технічні обмеження: [Вимоги до даних, очистки після тестів, безпека виведення з ладу середовищ, опис авторизації і доступів, як впливає тестування на учасників процесу, доступність інфраструктури, що неможливо виконати з точки зору технологій на проекті чи інструменту тощо].

Таблиця 1

Порівняння інструментів автоматизації тестування

Назва порівняльної характеристики	Опція 1 – Використання [назва інструменту 1] на [назва мови програмування]	Опція 2 – Використання [назва інструменту 2] на [назва мови програмування]
1	2	3
Огляд	<p>[Назва інструменту 1] – це потужне рішення для [типи тестування] з відкритим кодом. [Назва інструменту 1] розроблено для безперервного тестування [чого] та інтегрується з вашим конвеєром розробки. [Назва інструменту 1] має [перелік] і кольорові звіти.</p>	<p>[Назва інструменту 2] – сучасний інструмент для [типи тестування], створений для [опис]. [Назва мови програмування] – мова інтерфейсу. Код скомпільовано в [назва мови програмування]. Механізм [назва інструменту 2] написаний на [назва мови програмування], що робить його одним із найефективніших доступних інструментів тестування [типи тестування].</p>
Посилання	https://.../	https://.../
Переваги та недоліки	<ul style="list-style-type: none"> – безкоштовна версія надає необхідні речі для тестування – є досвід використання, але [опис] – легко писати тести – можна використовувати в конвеєрі (Pipelines/CI/CD) – stdout звіт у консолі конвеєра – дані з налаштування легко передаються в сценарії – візуалізація результатів, інше на Gitlab в артефактах – Звіт JUnit на Gitlab у звіті про тестування конвеєра – залежності [назва мови програмування] легко додаються через [назва збирача проекту] – відсутність порівняння результатів, можна імплементувати – складніше використання та з більшою кількістю рядків коду – безкоштовна версія не забезпечує розподілене навантаження з коробки, але це можна реалізувати 	<ul style="list-style-type: none"> – безкоштовна версія надає необхідні речі для тестування – легше писати тести – можна використовувати в конвеєрі (Pipelines/CI/CD) – комюніті реалізувала віджет для порівняння результатів – stdout звіт у консолі конвеєра – дані з налаштування легко передаються в сценарії – досвіду користування немає – не всі функції з коробки бібліотеки не можна використовувати безпосередньо, а як модулі – візуалізація результатів відсутня, може бути реалізована – Звіт JUnit відсутній, можна реалізувати безкоштовна версія не забезпечує розподілене навантаження з коробки, але це можна реалізувати – не всі додаткові запропоновані функції працюють належним чином і є недоробленими (наприклад, [приклад])

Продовження Таблиці 1

1		2	3	
Критерій	обов'язково	інструмент може бути застосовано у нашому випадку	так	так
	обов'язково	візуалізація результатів	підтримується з коробки	не надається, але є наступні варіанти: застосовувати [опис] ~ [оцінка зусиль] самостійна імплементація ~ [оцінка зусиль] застосувати розроблене спільнотою ~ [оцінка зусиль] також звіту stdout може бути достатньо для запуску автоматизації або як єдиний варіант
	було б добре мати	звіт JUnit	підтримується з коробки	не надається, але є наступні варіанти: самостійна імплементація ~ [оцінка необхідних зусиль], [фото] застосувати розроблене спільнотою ~ [оцінка необхідних зусиль], [фото]
	обов'язково	підтримує протоколи	відповідає проекту [назва проекту], [перелік протоколів]	відповідає проекту [назва проекту], [перелік протоколів]
	було б добре мати	можна розширити на інші проекти	так, без особливих зусиль [на всі або перелік проектів]	так, без особливих зусиль [на всі або перелік проектів]
	було б добре мати	можливість передачі параметрів запуску ззовні	так	так
	було б добре мати	можливість визначення набору тестів ззовні	так	так
	обов'язково	паралельний запуск тестів	так	так
	обов'язково	послідовне виконання окремих сценаріїв	так	так
	-	вхідний поріг	трохи вище варіанту 2, загалом 2-3 тижні з нуля або для новачків	загалом 2-3 тижні з нуля або для новачків
	-	простий у використанні	так, після ознайомлення або в разі copy-paste з наступним зануренням глибше	так, після ознайомлення або в разі copy-paste з наступним зануренням глибше
	-	щоденне використання	4 години в разі copy-paste без додаткових досліджень	4 години в разі copy-paste без додаткових досліджень

Продовження Таблиці 1

1		2		3	
було б добре мати	порівняння результатів	не передбачено, може бути реалізовано, не досліджено	так, але має плюси і мінуси [опис]		
не актуально зараз	великі навантаження	можливо, доведеться дослідити; наразі це [опис], [опис ресурсів]	можливо, доведеться дослідити; наразі це [опис], [опис ресурсів]		
не актуально зараз	розподілене навантаження	не передбачено для Open-Source, може бути реалізовано – приблизне рішення: запускати одночасно, збирати показники, генерувати консолідований звіт	не надається Open Source, може бути реалізовано – приблизне рішення: запускати одночасно, збирати показники, генерувати консолідований звіт		
обов'язково	можливість використання безкоштовної версії	безкоштовна версія надає необхідні речі для тестування	безкоштовна версія надає необхідні речі для тестування		
не актуально зараз	ціна	[опис ціноутворення, з урахуванням перспективи розширення], [посилання на ціни]	[опис ціноутворення, з урахуванням перспективи розширення], [посилання на ціни]		

Джерело: таблицю складено на базі власного успішного досвіду

Рекомендації.

Варіант 1 готовий до використання, оскільки візуалізація результатів і звітність JUnit надаються з коробки. Відсутність порівняння результатів не є критичною.

Варіант 2 немає візуалізації результатів із коробки, але може бути реалізовано. Також можна використовувати в майбутньому для інших проектів, це залежить від випадку.

Наслідки.

– у майбутньому може знадобитися дослідити, як налаштувати (стосується обох варіантів):

- високі навантаження;
- розподілене навантаження;

– для інших проектів може бути використаний інший фреймворк.

Що потрібно зробити.

Файли, посилання.

Висновки. Забезпечення якості (QA) ПЗ (та інших результатів господарювання) є частиною процесу управління якістю (TQM), що в свою чергу входить у цілісний продукт та вимагає прийняття відповідних ефективних управлінських рішень, які базуються на відповідному науковому підході. Загальновідомо, що автоматизація є найефективнішим інструментом забезпечення якості (AQA), який важливо при-

мінати при кожній можливості застосування. Щоб реалізувати на проекті, необхідно провести вибір тестового фреймворку, для чого найкраще підходить метод доведення концепції, що включає метод табличного порівняння альтернативних варіантів, для визначення і застосування найкращого. Для належного розуміння рекомендованої парної технології PoC-Benchmarking, високо-результативним є знайомство з різноманітними життєвими прикладами, один з яких наведено у цьому поданні, з прив'язкою до відповідної наукової термінології з економіки, менеджменту і IT, для розширеного бачення і наукового підходу в реальних умовах примінення.

На послідок варто зазначити базові моменти філософії економіки. Дешевші рішення можуть як правило в результаті призвести до більших витрат. Хто уникає використання серйозних методологій, той платить двічі, витрачає більше часу і втрачає темп. Це також призводить погіршення експертизи (спеціалізації), клімату і культури компанії, а також виснаження і демотивації людських ресурсів, втрати авторитету, іміджу та клієнтів, як наслідок – прибутків. Наука – це основа ефективності і упорядкування хаосу в господарюванні.

REFERENCES:

1. "Proof of Concept: From Idea to Implementation" by John Doe (2023) – Latest strategies and best practices for translating ideas into PoCs.
2. "Creating a Proof of Concept for Your Idea: A Practical Guide" by Susan McDonald (2021) – Practical guide to developing PoCs for new ideas.
3. "The Art of the PoC: A Guide to Building and Presenting Proof of Concepts" by Scott Ambler (2020) – Strategies for creating and presenting PoCs effectively.
4. "The PoC Methodology: A Guide to Building Successful Proof of Concepts" by Jeff Sutherland (2018) – Methodologies for developing and executing PoCs.
5. "Proof of Concept: A Practical Guide" by Michael J. McDonald (2014) – Practical advice on developing and implementing PoCs.
6. "Benchmarking: A Complete Guide to Best Practices" by Mary D. Campbell (2021) – Comprehensive guide to modern benchmarking practices.
7. "Benchmarking for Success: Optimizing Business Performance" by Lisa Robinson (2022) – Latest techniques and best practices in benchmarking.
8. "Business Benchmarking: A Practical Guide" by Paul J. F. Kelly (2012) – Practical insights into business benchmarking.
9. "ITIL® Foundation: ITIL 4 Edition" by Axelos (2019) – Updated framework for IT service management relevant to benchmarking.
10. "The Benchmarking Book: A How-to-guide to Best Practice for Managers and Practitioners" by Michael J. Spendolini (1992) – Foundational text still referenced for benchmarking practices.