ETUDE EXPLORATOIRE SUR L'IDENTIFICATION DES PRATIQUES DE REUTILISATION DES MICROSERVICES ASSOCIEES A LA MATURITE DEVOPS DECRITES DANS LA LITTERATURE ET LEURS OBSERVABILITES SUR LE TERRAIN

Mémoire présenté

dans le cadre du programme de maîtrise en informatique

en vue de l'obtention du grade de maître ès sciences (M. Sc.)

PAR

© ELIKEE WANDA MESSAN

Mai 2026

Composition du jury:

Mohamed Tarik Moutacalli, président du jury, UQAR

Mehdi Adda, directeur de recherche, UQAR

Hamid Mcheick, membre interne, UQAC

Dépôt initial le 15 janvier 2026 Dépôt final le 5 mai 2026

Université du Québec à Rimouski

Service de la bibliothèque

Avertissement

La diffusion de ce mémoire ou de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire « *Autorisation de reproduire et de diffuser un rapport, un mémoire ou une thèse* ». En signant ce formulaire, l'auteur concède à l'Université du Québec à Rimouski une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de son travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, l'auteur autorise l'Université du Québec à Rimouski à reproduire, diffuser, prêter, distribuer ou vendre des copies de son travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de la part de l'auteur à ses droits moraux ni à ses droits de propriété intellectuelle. Sauf entente contraire, l'auteur conserve la liberté de diffuser et de commercialiser ou non ce travail dont il possède un exemplaire.

Je dédie ce travail à ma famille, qui a su me donner du courage, me soutenir et m’accompagner.

REMERCIEMENTS

Je tiens à remercier tout particulièrement l'UQAR, pour m'avoir donné l'opportunité d'étudier dans un cadre de travail qui est l'un des meilleurs et de haute qualité. Cela a contribué fortement à mon avancement académique.

Je suis reconnaissante envers Mehdi Adda, directeur de recherche et à Gabriel Morais, collaborateur. J'ai eu vraiment la chance d'être encadrée par vous, votre amour du travail excellent, votre soutien moral m'ont boosté à donner le meilleur de moi. Le travail d'équipe a été un plaisir pour moi avec vous.

Je tiens également à remercier mes parents Mr Kodjo Messan et Mme Tina Vovor, mon mari Bertrand Dossou-Yovo, ma sœur Ella Messan, ma sœur de cœur Sandra Lamboni, notre marraine Muriel Badjema et aux autres membres de ma grande famille, pour leur amour inconditionnel, soutient moral, financier et spirituel.

RÉSUMÉ

L'architecture à base de microservices est une approche de développement qui promeut la création de petits services autonomes, communiquant par le biais d'une interface API bien définie, et qui vise à limiter les interactions quotidiennes. Elle est souvent associée aux pratiques DevOps, une approche qui favorise la collaboration entre développeurs et opérateurs grâce à l'amélioration continue et une livraison plus fréquente des produits logiciels. Bien que ces deux approches soient complémentaires, de nombreuses entreprises se trouvent encore en phase de transition, ce qui entraîne des niveaux d'adoption variables.

L'objectif de cette étude est d'identifier si les pratiques de réutilisation des microservices associées à DevOps, telles que présentées dans la littérature, correspondent à celles observables en pratique. Pour atteindre cet objectif, nous répondons aux trois questions de recherche suivantes :

1. Quelles pratiques de réutilisation des microservices sont décrites dans la littérature et observées sur le terrain dans un contexte DevOps?
2. Quels sont les facilitateurs de la réutilisation identifiés dans la littérature et perçus par les professionnels interrogés ?
3. Quelles barrières à la réutilisation sont rapportées dans la littérature et identifiées dans la pratique?

Nous adoptons une méthodologie de recherche par sondage pour répondre à ces questions. Ce travail s'appuie, d'une part, sur une revue de littérature permettant d'identifier les pratiques de réutilisation des microservices associées à la maturité DevOps décrites dans les travaux scientifiques et, d'autre part, sur une enquête menée auprès de dix-sept professionnels repartis dans deux régions différentes : l'Afrique de l'Ouest francophone (Senegal, Togo) et le Québec (Canada). Les résultats obtenus permettent de comparer les pratiques rapportées dans la littérature avec celles réellement mises en œuvre dans les organisations. Ils offrent un aperçu concret des pratiques actuelles, des défis rencontrés, ainsi que des conditions favorables ou défavorables à la réutilisation dans divers contextes organisationnels.

Mots clés : Architecture à base de microservices, DevOps, réutilisation logicielle, qualité logicielle, modèle de qualité, attribut de qualité, recherche par sondage.

ABSTRACT

Microservices architecture is a development approach that promotes the creation of small, autonomous services communicating through a well-defined API interface, aiming to minimize daily interactions. It is often associated with DevOps practices, which foster collaboration between developers and operations teams through continuous improvement and more frequent software product delivery. While these two approaches are complementary, many companies are still in a transition phase, resulting in varying levels of adoption.

The objective of this study is to determine whether the microservices reuse practices associated with DevOps, as presented in the literature, correspond to those observed in practice. To achieve this objective, we address the following three research questions:

1. What microservices reuse practices are described in the literature and observed in practice within a DevOps context?

2. What facilitators of reuse are identified in the literature and perceived by the professionals interviewed?

3. What barriers to reuse are reported in the literature and identified in practice?

To answer these questions, we adopted a survey research methodology. This work is based, firstly, on a literature review to identify microservices reuse practices associated with DevOps maturity as described in the scientific literature and, secondly, on a survey of seventeen professionals from two different regions: French-speaking West Africa (Senegal, Togo) and Quebec (Canada). The results obtained allow us to compare the practices reported in the literature with those implemented in organizations. They offer a concrete overview of current practices, the challenges encountered, and the conditions that are favorable or unfavorable to reuse in various organizational contexts.

Keywords: Microservices architecture, DevOps, software reuse, software quality, quality model, quality attribute, survey research.

TABLE DES MATIÈRES

REMERCIEMENTS IX

RÉSUMÉ XI

ABSTRACT XIII

TABLE DES MATIÈRES XV

LISTE DES TABLEAUX XVIII

LISTE DES FIGURES XX

LISTES DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES XXII

INTRODUCTION GÉNÉRALE 1

1. CONTEXTE 1
2. PROBLEMATIQUE 2
3. OBJECTIFS 2
4. METHODOLOGIE 3
5. CONTRIBUTIONS 4
6. ORGANISATION DU MEMOIRE 4

CHAPITRE 1 ETAT DE L'ART 5

1. STRATEGIE DE RECHERCHE DOCUMENTAIRE 5
2. CONTEXTE THEORIQUE 6
   - 2.1 Qualité logicielle 7
   - 2.2 Attribut et modèle de qualité 8
   - 2.3 Réutilisation logicielle 12
   - 2.4 Architecture à base de microservices 13
   - 2.5 DevOps 15
   - 2.6 Interdépendance entre qualité logicielle, architecture à base de microservices et DevOps 15
3. TRAVAUX CONNEXES 16
   - 3.1 MSA et DevOps dans la littérature 16
   - 3.2 Réutilisation logicielle 17

CHAPITRE 2 METHODOLOGIE 21

1. CONCEPTION ET STRUCTURATION DU QUESTIONNAIRE 22
2. SELECTION DES PARTICIPANTS 27
3. LA DIFFUSION DU QUESTIONNAIRE ET SUIVI DE REPONSES 27
4. METHODE DE COLLECTE ET D'ANALYSE DE DONNEES 28

CHAPITRE 3 29

EXPLORATION DES PRATIQUES DE REUTILISATION DES MICROSERVICES DANS UN CONTEXTE DEVOPS 29

CONCLUSION GÉNÉRALE 44

1. CONTRIBUTION A LA RECHERCHE 45
2. LIMITES DE L'ETUDE 45
3. PERSPECTIVE ET RECOMMANDATION 46

ANNEXE : 49

PRESENTATION DU QUESTIONNAIRE 49

REFERENCES BIBLIOGRAPHIQUES 58

LISTE DES TABLEAUX

Tableau 1: Synthèse comparative des modèles de qualités et d'évaluation dans le contexte MSA/DevOps 10
Tableau 2: Correspondance entre les concepts théorique de la réutilisation et des questions du formulaire de notre enquête. 24
Tableau 3: Correspondance des questions du formulaire de notre enquête avec Microservices et DevOps. 25

LISTE DES FIGURES

Figure 1: Workflow méthodologique de l'étude, adapté des recommandations de Lin et de Mell 21
Figure 2. Secteurs de travail des participants 37
Figure 3. Microservices par secteur 37

LISTES DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

MSA Architecture à base de microservices

DevOps Développement et Operations

CI/CD Intégration continue / Déploiement continus

SOA Architecture orienté service

API Interface de programmation applicative

INTRODUCTION GÉNÉRALE

1. CONTEXTE

L'architecture à base de microservices, en tant que style architectural moderne, suscite un intérêt croissant tant dans l'industrie que dans le monde académique (Zimmermann, 2017) ;(Newman, 2015). MSA a été largement adoptée par l'industrie dans différents domaines afin de résoudre les problèmes apportés par l'architecture orientée services (SOA) et monolithique (Cojocaru et al., 2019). Les microservices ont émergé des pratiques adoptées par des géants du web, tels qu'Amazon, Netflix, Spotify, qui ont senti le besoin d'innover rapidement, de modifier ou de déployer individuellement des services indépendamment du reste du système (Newman, 2015). L'architecture à base de microservices est une approche de développement d'applications en tant que suite de petits services s'exécutant chacun dans son propre processus et communiquant avec des mécanismes légers (Di Francesco et al., 2019) ; (Richards, 2015).

L'adoption de l'architecture à base de microservices offre aux organisations une flexibilité technologique, une plus grande évolutivité, une livraison plus rapide, un déploiement facilité grâce au découplage qui réduit les dépendances entre équipes, leur permettant ainsi de travailler de manière autonome sur des services modulaires où chaque service peut être déployé sans affecter tout le système et chaque équipe est responsable d'un microservice, soutenu par des interfaces API bien définies qui limitent des interactions quotidiennes. (Zimmermann, 2017) ; (Dragoni et al., 2017) ; (Richards, 2015).

En parallèle, DevOps est une méthodologie visant à combler des fossés, tels que la différence de responsabilité, différences de rythme de travail, problèmes de communication entre le développement et les opérations, en mettant l'accent sur la communication, la collaboration, l'intégration continue, l'assurance qualité et la livraison (Jabbari et al., 2016). Poussées par le besoin de répondre rapidement aux exigences des clients, de nombreuses entreprises ont adopté cette méthode. DevOps et l'architecture à base de microservices sont deux approches de développement qui partagent plusieurs principes : amélioration continue, automatisation des processus, livraison continue et

assurance qualité des produits. Ainsi, l'intégration de ces deux concepts suscite un intérêt croissant et souligne l'influence mutuelle entre ces deux pratiques (Waseem et al., 2020) ; (Bogner et al., 2019).

2. PROBLEMATIQUE

La réutilisation des logiciels est un principe clé du génie logiciel, présent depuis de nombreuses années, et qui consiste à utiliser des artefacts existants pour développer de nouveaux systèmes logiciels au lieu de partir de zéro. Elle constitue une pratique fiable et rentable pour la création de grands systèmes logiciels (Morais, 2023). Bien que la réutilisation soit une pratique ancienne du génie logiciel, dans les contextes MSA/DevOps, elle peut être affectée par plusieurs facteurs :

- Les principes d'indépendance des services, d'où l'autonomie des services soutenue par « l'architecture sans partage » des microservices où chaque service possède et gère ses propres ressources ; et du principe de collaboration du DevOps.
- La modularité des microservices favorise la réutilisation en facilitant la consommation des services, qui peuvent être consommés librement sans validation du fournisseur tant qu'il respecte le contrat d'API ce qui permet d'améliorer la qualité et de réduire les risques de défauts puisqu'ils ont déjà été testés, déployés et stabilisés.
- La pratique DevOps fondée sur l'automatisation, rend les processus de construction et de déploiement reproductibles tout en facilitant la réutilisation des chaînes d'intégration, puisqu'ils sont définis dans un format de code applicatif (Morais et al., 2024).

Cependant, malgré ses avantages, la réutilisation dans le contexte des microservices rencontre des défis spécifiques liés à la documentation, à la coordination et à la maintenance (Badampudi et al., 2024).

3. OBJECTIFS

L'objectif de cette étude est d'identifier les pratiques de réutilisation des microservices en contexte DevOps et d'examiner leur correspondance entre littérature et pratique.

Pour ce faire, trois questions orientent notre recherche :

- Quelles pratiques de réutilisation des microservices sont décrites dans la littérature et observées sur le terrain dans un contexte DevOps ?
- Quels sont les facilitateurs de la réutilisation identifiés dans la littérature et perçus par les professionnels interrogés ?
- Quelles barrières à la réutilisation sont rapportées dans la littérature et identifiées dans la pratique ?

4. METHODOLOGIE

Notre démarche combine :

- Une revue de littérature, qui nous a permis de comprendre les concepts clés et fondamentaux de notre étude, tels que microservices, assurance qualité, attribut de qualité, modèle de qualité et de DevOps. Elle nous a permis également une mise à jour des études récentes portant sur les MSA.
- Une enquête de type sondage, conçue en suivant les recommandations méthodologiques de (Linåker et al., 2015) pour les études par questionnaire en génie logiciel. Le questionnaire a été construit à partir de l'étude de trois modèles de qualité, modèle de qualité de maturité DevOps, modèle de réutilisation et modèle de DeLone et McLean proposée respectivement par ((Chen, 2018) ; (W. Frakes & Terry, 1996) ; (Mardiana et al., 2015)), combinée à la littérature et basée sur nos connaissances.

Le questionnaire, évalué par des experts, a été envoyée par courriel à nos participants de l'Afrique francophone (Senegal, Togo) et le Québec (Canada). Nous avons ciblé ces terrains d'étude, car c'est dans ces régions que nous avions accès à plus de praticiens impliqués dans des projets microservices, notamment en raison de nos réseaux personnels. De plus, ces deux régions présentent des niveaux de maturité technologique différents, permettant d'examiner l'adoption et les pratiques de réutilisation dans des environnements différents.

Nous avons obtenu un total de 17 réponses sur les 31 demandes envoyées. Les sollicitations étaient réparties entre Sénégal, Togo (Afrique de l'ouest) et le Quebec (Canada). La collecte a été réalisée grâce à l'outil Google Forms. Après application des critères d'inclusion, exigeant que le répondant soit impliqué dans un projet utilisant une architecture à base de microservices, de manière actuelle ou planifiée, et des critères d'exclusion éliminant les réponses incohérentes ou incomplètes, 14 réponses ont été retenues pour notre analyse. Nous avons calculé la fréquence, le mode et réalisé un tableau de croisement, comme recommandé pour les enquêtes exploratoires à échelles nominales à petit échantillon.

5. CONTRIBUTIONS

Les résultats de cette recherche mettent en parallèle les pratiques rapportées dans la littérature et celles observées en entreprise. Elle fournit des données empiriques nouvelles sur la réutilisation en MSA/DevOps et met en lumière les défis réels et les conditions favorables ou défavorables, tout en comparant deux contextes technologiques différents.

6. ORGANISATION DU MEMOIRE

Le mémoire est structuré comme suit : le premier chapitre présente l'état de l'art et définit en détail les concepts clés résumant notre étude. Le deuxième chapitre décrit la méthodologie adoptée. Le troisième chapitre présente l'article scientifique issu du projet. Enfin, une conclusion générale synthétise les résultats de l'étude, discute de sa portée et de ses limites et propose des perspectives ainsi que des recommandations pour de futures recherches.

CHAPITRE 1
ETAT DE L'ART

1. STRATEGIE DE RECHERCHE DOCUMENTAIRE

Nous avons d'abord pris le temps de nous familiariser avec les concepts de base et essentiels à notre sujet : microservices, assurance qualité, attribut de qualité, modèle de qualité, DevOps. À partir d'une recherche exploratoire visant à évaluer l'état actuel de l'adoption des microservices (MSA) dans l'industrie, nous avons étudié des modèles de qualité des microservices qui nous ont permis d'identifier les attributs de qualité spécifiques au MSA. Cette première analyse nous a permis d'acquérir une compréhension approfondie du sujet et d'identifier les principaux axes d'étude des MSA. Les travaux récents portaient principalement sur l'influence mutuelle entre MSA et DevOps, de la réutilisation des microservices, d'une possibilité que la réutilisation constitue un attribut de qualité commun à ces deux approches. Les études recensées soulignent également les défis, avantages et obstacles associés a la réutilisation des MSA et du DevOps.

De ces constats, il nous a semblé judicieux d'orienter notre étude vers l'examen des pratiques réelles sur le terrain. Ainsi, l'objectif de notre étude est : « d'identifier dans quelle mesure les pratiques de réutilisation des microservices décrites dans la littérature et associées à la maturité DevOps correspondent à celles observées sur le terrain ». Pour répondre à cet objectif, nous avons choisi de faire une enquête auprès de deux régions ayant une situation géographique différente, permettant une comparaison de contextes technologiques différents.

La conception de notre enquête s'est appuyée sur trois (3) modèles de qualité : le modèle de maturité, qui nous a permis de comprendre au mieux les étapes et pratiques essentielles pour adopter et améliorer DevOps ; le modèle de réutilisation nous a permis de connaître les conditions et de comprendre le mécanisme rodant autour de la réutilisation des microservices et le modèle DeLone et McLean nous a servi de référence pour prendre en compte des critères de qualité que doit respecter notre formulaire lors de

son élaboration. Un formulaire a été fait pour notre enquête, garantissant ainsi des résultats en adéquation avec notre problématique.

Enfin, une analyse claire et bien structurée a été réalisée grâce à la collecte et à la manipulation des données.

Sélection des articles : Afin d'assurer une revue de littérature rigoureuse, nous avons défini une stratégie de recherche basée sur des mots-clés spécifiques : « Microservices », « Microservices Survey » et « Microservices AND DevOps », « Microservices AND Reuse ». Ces termes ont été sélectionnés pour maximiser la pertinence des études collectées et couvrir un large éventail de travaux académiques traitant des microservices et de leur adoption dans l'industrie. Les recherches ont été effectuées sur des bases de données académiques reconnues, notamment IEEE Xplore, google scholar, scopus, researchGate et sciencedirect, afin d'accéder à des publications de qualité et validées par des pairs. Notre premier critère d'inclusion était la nature des études. Étant donné que notre travail s'inscrit dans une étude par sondage, seuls les articles relevant d'une revue de littérature ou d'une étude empirique de type enquête ont été retenus. Les travaux ne correspondant pas à ces critères ont été écartés.

Affinement de la sélection : Parmi les articles présélectionnés, nous avons appliqué des critères supplémentaires pour affiner notre corpus :

- Année de publication : nous avons privilégié les travaux récents afin d'assurer une actualité des informations.
- Titre et résumé : l'article devait explicitement mentionner une analyse des microservices et/ou DevOps en lien avec notre thématique.
- Objectifs de l'étude : nous avons sélectionné les travaux dont les objectifs étaient alignés sur notre problématique, notamment ceux portant sur les défis, les meilleures pratiques et l'impact de la réutilisation des microservices dans l'industrie, et ces impacts sur DevOps.

2. CONTEXTE THEORIQUE

2.1 Qualité logicielle

La qualité d'un logiciel se mesure par sa capacité à répondre aux besoins des clients tout en respectant les bonnes pratiques de développement (Estdale & Georgiadou, 2018). Plusieurs aspects peuvent être considérés pour en évaluer la qualité :

- La documentation et les tests : la documentation explique comment utiliser le logiciel, tandis que les tests permettent de vérifier son bon fonctionnement.
- Les processus de développement : ils regroupent les méthodes et pratiques employées pour la création du logiciel.
- Les composants du logiciel : cela inclut le code, les fonctionnalités et l'architecture du système.

La qualité peut être abordée sous différents angles, la qualité interne et la qualité externe :

- La qualité externe : correspond à la satisfaction des utilisateurs finaux. Elle se manifeste uniquement lors de l'utilisation du logiciel et prend en compte des critères tels que la performance, la fiabilité, la facilité d'utilisation et la productivité (Carvallo & Franch, 2006).
- La qualité interne : concerne les aspects liés au développement du produit, notamment ses caractéristiques techniques et sa maintenabilité. Elle est mesurée tout au long du processus de création (Carvallo & Franch, 2006).

Dans notre étude nous nous focalisons sur la qualité interne dans le contexte des microservices. Pour évaluer pleinement la qualité interne des microservices, il convient de s'appuyer sur deux notions essentielles : les attributs de qualité et les modèles de qualité.

2.2 Attribut et modèle de qualité

Un attribut de qualité est un ensemble de facteurs communs qui affectent un système lors de son exécution.

D'un point de vue technique, les attributs de qualité déterminent d'importantes décisions en matière d'architecture et de conception (Tapia & Gaona, 2023). Mesurer la qualité d'un logiciel revient à vérifier s'il respecte les objectifs définis au départ et si le processus de développement répond aux besoins du client (Estdale & Georgiadou, 2018). Pour cela, différents attributs sont considérés, tels que : la réutilisation (capacité à réutiliser les modules développés) ; la fiabilité (la capacité d'un logiciel à exécuter ses fonctions prévues sans défaillance) ; la résilience (aptitude du logiciel à continuer de fonctionner malgré les défaillances de certains composants). Dans notre étude, seule la réutilisation nous intéresse en tant qu'attribut de qualité pour mesurer la qualité des logiciels MSA.

Pour évaluer ces attributs, nous avons mis en place des modèles de qualité. Plusieurs modèles de qualité ont été proposés, tels que : le modèle de McCall, qui a pour but de mesurer les caractéristiques de haut niveau, le modèle de Boehm qui propose une vue plus élargie avec une collection variée de caractéristiques ; la norme ISO/IEC 25000, il fournit le modèle d'architecture Square (System and Software Quality Requirements and Evaluation), un cadre structuré pour évaluer la qualité des produits logiciels ; le modèle de maturité DevOps pour évaluer la maturité des systèmes DevOps et le modèle de réutilisation qui permet d'identifier les conditions de réutilisation . Le modèle de McCall vise à mesurer des caractéristiques de haut niveau (facteurs de qualité) ainsi que des critères plus spécifiques (critères de qualité).

Le modèle de Boehm propose une vue plus élargie avec une collection variée de caractéristiques ; la norme ISO/IEC 25000 fournit le modèle d'architecture Square (System and Software Quality Requirements and Evaluation), un cadre structuré pour évaluer la qualité des produits logiciels.

Le modèle de maturité DevOps proposé par (Chen, 2018), évalué le degré d'intégration, d'automatisation et de maîtrise des pratiques DevOps au sein d'une

organisation. Il se manifeste par la capacité d'une équipe à livrer fréquemment, de manière fiable et avec un haut degré d'autonomie. C'est un modèle conceptuel qui se repose sur trois dimensions interdépendantes : l'architecture à base de microservices, qui permet des déploiements indépendants et favorise l'autonomie des équipes ; la livraison continue, qui repose sur l'automatisation des tests, des déploiements et sur une rétroaction rapide ; les pratiques DevOps, qui instaurent une culture de collaboration, de responsabilité partagée et d'amélioration continue. Le modèle de maturité DevOps selon (Chen, 2018) montre que la maturité n'est pas uniquement organisationnelle : elle dépend fortement de l'architecture logicielle et du niveau d'automatisation opérationnelle qui la soutient.

Le modèle DeLone McLean est un modèle qui propose un cadre pour évaluer le succès d'un système d'information aux dépens de quatre types de qualité en me référant qu'en ses dimensions en termes de qualité : qualité du système, qualité de l'information produite, la qualité de son utilisation, la qualité du service (Mardiana et al., 2015). Dans notre étude, l'utilisation de tous ces modèles nous a permis de garantir la qualité et la rigueur de notre enquête.

Dans le cadre de la conception de notre questionnaire, nous avons principalement retenu les modèles suivants : Le modèle de maturité DevOps accompagne la croissance des organisations productrices de logiciels (Chen, 2018) ; le modèle de réutilisation qui permet d'identifier et de réutiliser des composants logiciels dans différents projets (W. Frakes & Terry, 1996) ; le modèle DeLone McLean (Mardiana et al., 2015). Le tableau 1 compare les modèles de qualité retenus et précise leur utilité dans un contexte microservices/DevOps. Cette comparaison met aussi en évidence certaines limites de ces modèles face aux exigences particulières des environnements MSA/DevOps, notamment en termes de rapidité de livraison, de granularité des services et d'automatisation des processus.

Tableau 1: Synthèse comparative des modèles de qualités et d'évaluation dans le contexte MSA/DevOps

Modèle de qualité	**Type de modèles**	**Méthodes/Mode d'évaluation**	**Avantages dans un contexte MSA/DevOps**	**Limites dans un contexte MSA/DevOps**	**Usage dans l'étude**	**Références**
Modèle de McCall	Modèle initial	Critères et métrique	Décomposition fine des qualités internes	Peu adapté aux interactions légères entre services	Identifier et mesurer les facteurs de qualité	(McCall et al., 1977)
Modèle de Boehm	Modèle standard	Approche descendante	Vision holistique incluant satisfaction client	Complexité pour de nombreux microservices	Reference pour définir la qualité logicielle	(Boehm, 1978)
Modèle ISO/IEC 25000 (Sqaure)	Norme	Cadre structuré	Standardisation moderne de la qualité	Rigidité face à la rapidité DevOps	Cadre structurant pour l'évaluation	(Estdale & Georgiadou , 2018)

Modèle de Qualités DevOps	Modèle de maturité	Évaluation des processus	Mesure de l'automatisation et culture DevOps	Peu centré sur les attributs techniques	Elaboration des questions sur l'automatisation	(Chen, 2018)
Modèle de réutilisation	Modèle oriente composants	Condition de réutilisation	Identification précise des opportunistes	Limité par l'architecture sans partage	Evaluer pratiques systématiques/opportunistes	(W. Frakes & Terry, 1996)
Modèle DeLone et McLean	Modèle de succès SI	Dimensions qualité/usage/satisfaction	Intègre la perception utilisateur	N'évalue pas les attributs techniques	Base pour la conception du questionnaire.	(Mardiana et al., 2015)

2.3 Réutilisation logicielle

La réutilisation des logiciels est l'utilisation de logiciels existants ou de connaissances en logiciels pour construire de nouveaux logiciels (W. B. Frakes & Kang, 2005). Elle a pour but de réduire les coûts, le délai de développement et d'améliorer la qualité.

On a deux types de réutilisation :

- **Réutilisation systématique :** c'est une approche institutionnalisée intégrée dans la stratégie de développement de l'organisation, qui met à disposition des outils et standards afin de garantir que les composants soient réutilisables (W. B. Frakes & Kang, 2005) ;
- **Réutilisation opportuniste** : c'est une approche dans laquelle on développe de nouveaux systèmes logiciels en réutilisant et en combinant les composants qui n'ont pas été conçus pour être utilisés ensemble (Mäkitalo et al., 2020).

Dans le contexte de l'architecture à base de microservices, la réutilisation présente des spécificités particulières : d'une part, elle est favorisée par la modularité et le découplage des services qui peuvent être consommés librement tant qu'ils respectent leur contrat d'interface API. D'autre part, elle est confrontée à des défis importants liés au principe de l'architecture sans partage ou encore le principe d'indépendance, selon lequel chaque service doit gérer ces propres ressources pour rester autonomes. Ce principe limite la réutilisation directe des ressources tout en encourageant une réutilisation fondée sur la consommation des ressources.

Dans un environnement DevOps, la réutilisation est également favorisée par l'automatisation des processus, notamment à travers les chaînes d'intégration, les scripts d'infrastructure et les configurations qui deviennent eux-mêmes des artefacts réutilisables et reproductibles. Cependant, malgré ses bénéfices, la réutilisation en contexte MSA/DevOps demeure complexe. Des études récentes, telles que celle de (Badampudi et al., 2024) , mettent en évidence des défis spécifiques liés à la documentation, a la

coordination entre équipes et la maintenance. Cette étude souligne également que l'approche InnerSource peut favoriser la réutilisation à grande échelle, au prix d'efforts initiaux de standardisation.

Le paradoxe important ressort des travaux existants : les microservices reposent sur un principe d'indépendance forte (architecture sans partage), alors que la réutilisation implique une forme de dépendance ou de partage. Les auteurs réconcilient ce paradoxe en distinguant plusieurs niveaux de réutilisation. La réutilisation ne porte généralement pas sur les ressources internes d'un service, mais sur des artefacts périphériques ou contractuels.

Ainsi, la littérature identifie plusieurs formes de réutilisation compatibles avec les principes MSA :

1. La réutilisation de templates ou squelettes de services permettant de standardiser la création de nouveaux microservices ;
2. L'usage de bibliothèques communes ou SDK, bien que leur emploi puisse introduire un couplage indésirable ;
3. La réutilisation d'API ou de capacités exposées par d'autres services, qui constitue la forme la plus conforme aux principes d'indépendance ;
4. La réutilisation d'artefacts DevOps tels que les chaînes d'intégration CI/CD, les scripts d'infrastructure ou les configurations reproductibles.

La littérature montre ainsi que la réutilisation dans les microservices se déplace du code interne vers des interfaces, des patterns et des artefacts d'automatisation, permettant de préserver l'autonomie des services tout en favorisant la standardisation

2.4 Architecture à base de microservices

Les microservices sont une suite de services granulaires qui peuvent être développés, testés et déployés sur différentes plateformes à travers plusieurs technologiques (Larrucea et al., 2018). Contrairement aux microservices, on a les

systèmes monolithiques qui recommandent en retour, des applications unifiées en un seul bloc. Cela rend les modifications plus complexes (Richards, 2015).

L'architecture à base de microservices est une architecture qui préconise de décomposer une application en de petits services simples et indépendants les uns des autres, en les communiquant entre eux par le biais de mécanismes légers (API RESTful). Chacun de ces services est responsable d'une fonctionnalité spécifique pouvant être déployée et maintenue de manière autonome (Newman, 2015) ;(Richards, 2015). L'architecture à base de microservices présente de nombreux avantages en termes de qualité logicielle, tels que :

- Amélioration de la maintenabilité et de la testabilité : qui est dû à la création des modules plus petits et des équipes expertes dédiées ;
- Amélioration de la productivité : pour alléger le fardeau des IDE, pour une meilleure compréhensibilité et une intégration plus facile ;
- Isolement d'anomalies : vu la fonctionnalité spécifique de chaque service, une panne dans un service n'affectera pas les autres services ;
- Flexibilité et évolutivité **:** élimination de l'engagement à long terme envers les technologies, plus grande évolutivité et une livraison rapide.

Les avantages offerts par les MSA s'accompagnent de défis importants tels que la complexité en matière de gestion des dépendances, de surveillance, de débogage et de gouvernance liés à la création de plusieurs services. En raison de l'autonomie fonctionnelle de chaque service et de la répartition des responsabilités, la cohérence des données devient également un enjeu central dans les MSA. Pour résoudre ces défis, les organismes s'appuient de plus en plus sur des pratiques d'automatisation, de surveillance et de coordination entre équipes, qui sont étroitement associées à l'approche DevOps (Waseem et al., 2021). Ainsi MSA et DevOps partagent des principes fondamentaux tels que l'automatisation des processus, l'amélioration continue et la livraison continue. Toutefois, on remarque une différence entre la vision idéale et la réalité sur le terrain, en particulier à cause de la complexité accrue de la gestion et de la surveillance des systèmes (Zhou et al., 2023) ; (Tanzil et al., 2023).

2.5 DevOps

DevOps est une approche de développement collaboratif et multidisciplinaire au sein d'une organisation dont l'adoption vise à briser les barrières traditionnelles entre les équipes de développement (Dev) et d'exploitation (Ops), afin de favoriser le processus de livraison (Leite et al., 2019). DevOps a vu le jour grâce à l'essor de la méthode agile et du flux de travail continu et dont l'adoption permet l'amélioration et l'évolution des produits à une fréquence plus élevée par rapport aux processus de développement traditionnels des logiciels (Team, 2021). DevOps offre de nombreux avantages tels que : la collaboration et la culture organisationnelle, l'amélioration de la qualité logicielle, l'accélération des cycles de développement et de livraison, permettant aux organismes de livrer des fonctionnalités de manière plus fréquente et plus rapide (Leite et al., 2019).

Dans le contexte microservices, DevOps permet d'améliorer la maintenabilité et l'évolutivité des systèmes grâce à des pratiques d'amélioration continue, de documentation et de la gestion des services. Ce qui favorise une mise à jour fréquente et sécurisée des services indépendants, tout en maintenant une stabilité entière du système (Zimmermann, 2017). Ces deux approches sont reconnues pour leurs complémentarité, les caractéristiques des microservices telles qu'un couplage faible, l'autonomie entre services et le déploiement indépendant s'observent également avec les principes DevOps, qui permettent une excellente agilité organisationnelle et technique(Dragoni et al., 2017) (Newman, 2015).

2.6 Interdépendance entre qualité logicielle, architecture à base de microservices et DevOps

La qualité logicielle constitue un cadre conceptuel permettant d'évaluer un produit logiciel à travers des attributs mesurables tels que la maintenabilité, la fiabilité ou la réutilisation, conformément aux modèles de qualité établis comme l'ISO/IEC 25010 (Estdale & Georgiadou, 2018). Parmi ces attributs, la réutilisation occupe une place stratégique dans les architectures contemporaines, car elle contribue à réduire les coûts

de développement et à accélérer les cycles de livraison (Dragoni et al., 2017) ;(Estdale & Georgiadou, 2018). L'architecture à base de microservices s'inscrit dans cette dynamique en décomposant le système en services indépendants, favorisant ainsi la modularité et la réutilisation de composants dans divers contextes (Newman, 2015); (Dragoni et al., 2017). Toutefois, cette granularité introduit également des défis techniques significatifs. La multiplication des services accroît les risques de régression, génère des incohérences potentielles entre versions et alourdit la charge opérationnelle liée à la coordination, au déploiement et au maintien de la qualité interne du système. Ces difficultés sont largement documentées dans la littérature sur les architectures distribuées, qui souligne que la gestion d'un ensemble de services autonomes nécessite des mécanismes robustes de synchronisation, de test et d'observabilité (Dragoni et al., 2017) ;(Newman, 2015). Dans ce contexte, l'approche DevOps joue un rôle déterminant en intégrant l'automatisation (CI/CD), l'Infrastructure as Code, le monitoring continu et une culture de collaboration, DevOps permet de maîtriser la complexité opérationnelle des systèmes distribués et de préserver la qualité globale du logiciel (Humble & Farley, 2010); (Bass et al., 2015); (Kim et al., 2021).

3. TRAVAUX CONNEXES

3.1 MSA et DevOps dans la littérature

Les travaux portant sur les microservices (MSA) et DevOps montrent que ces deux approches évoluent conjointement et s'influencent mutuellement. Plusieurs auteurs soulignent d'abord que les microservices constituent un levier important pour l'adoption de DevOps. (Bass et al., 2015) indiquent que la granularité des microservices facilite la mise en place de petites équipes autonomes, ce qui correspond directement aux principes DevOps. Dans la même perspective, (Chen, 2018) rappelle que l'architecture logicielle peut devenir un obstacle majeur à l'implémentation de DevOps lorsque les systèmes sont trop monolithiques ou insuffisamment modulaires. Les travaux plus conceptuels, comme ceux de (Leite et al., 2019) mettent en lumière la relation structurelle entre les MSA et DevOps. Ils montrent que la modularité, l'automatisation et la supervision distribuée soutiennent les pratiques DevOps, tout en complétant les constats de Bogner et Carvalho

sur les conditions d'implémentation. (Taibi et al., 2019) prolongent cette réflexion en identifiant les techniques microservices pouvant être intégrées au chaîne d'intégration DevOps, mais soulignent également des lacunes persistantes dans les pratiques industrielles.

À partir de 2021, la littérature se concentre davantage sur les défis organisationnels et opérationnels. (Waseem et al., 2021) montrent que DevOps est largement adopté pour améliorer la qualité, la fréquence des livraisons et la collaboration interdisciplinaire. Toutefois, leur étude met en évidence des obstacles liés à la culture organisationnelle, à la coordination entre équipes et à l'intégration avec les architectures existantes. (Li et al., 2021), à travers une revue de littérature sur l'assurance qualité dans les systèmes microservices, identifient six attributs de qualité clés et dix-neuf tactiques d'amélioration, confirmant que la qualité dans les MSA dépend autant des pratiques DevOps que des choix architecturaux. Les travaux de 2023 renforcent cette perspective. (Zhou et al., 2023) montrent que, malgré les bénéfices perçus des microservices, les organisations font face à une complexité accrue en matière de gestion, de surveillance et de coordination. (Tanzil et al., 2023) identifient 23 préoccupations techniques et organisationnelles rencontrées par les professionnels DevOps, confirmant que les obstacles ne sont pas uniquement technologiques, mais également culturels et structurels. (Azad & Hyrynsalmi, 2023) synthétisent ces constats en identifiant sept facteurs de succès des projets DevOps, répartis entre dimensions techniques, organisationnelles, sociales et culturelles.

Enfin, El et al. (2024) montrent que la convergence entre Agile, Cloud et DevOps crée à la fois des synergies (accélération des tests, automatisation accrue) et des tensions (responsabilités floues, standardisation insuffisante). Dans l'ensemble, ces travaux indiquent que MSA et DevOps se renforcent mutuellement, mais que leur intégration dépend fortement de la maturité organisationnelle, de la gouvernance et de la capacité des équipes à adopter des pratiques cohérentes.

3.2 Réutilisation logicielle

La réutilisation logicielle dans les architectures à base de microservices reste un thème encore peu documenté dans la littérature. Les premières études, comme celle de Carvalho et al. (2019), se concentrent sur l'extraction de microservices réutilisables à partir de systèmes existants. Elles montrent que la variabilité et la simplicité des mécanismes de configuration jouent un rôle central dans la réutilisation, mais restent centrées sur les aspects techniques de la transition vers les MSA. Les travaux de 2023 élargissent cette perspective. Hamed (2023) montre que la transformation d'un système monolithique en architecture à base de microservices peut renforcer la réutilisation, à condition de disposer de règles de transformation claires et centrées sur les fonctionnalités. Morais (2023), quant à lui, aborde explicitement la réutilisation dans un contexte DevOps. Il met en évidence l'absence de processus structurés, de standardisation et de gouvernance, et propose un modèle de référence pour guider les équipes dans la conception de microservices modulaires et réutilisables. Ces travaux soulignent que la réutilisation ne peut émerger spontanément : elle nécessite un cadre organisationnel et des pratiques cohérentes. Les études de 2025 adoptent une perspective plus systémique. Sr. et al. (2025) proposent un cadre visant à améliorer la réutilisation en traitant des défis tels que la duplication de code, l'hétérogénéité technologique, la gestion des versions et les décisions architecturales. Ce cadre offre une vision holistique, mais nécessite encore une validation empirique. Badampudi et al. (2025), à travers une étude de cas chez Ericsson, montrent que l'adoption d'InnerSource et de pratiques CI/CD peut favoriser la réutilisation à grande échelle, mais au prix d'investissements initiaux importants en matière de conformité, de standardisation et d'automatisation.

La littérature montre que les architectures à base de microservices et les pratiques DevOps évoluent ensemble et se renforcent mutuellement. Les études soulignent des bénéfices attendus (modularité, automatisation, autonomie des équipes) mais mettent aussi en évidence des défis récurrents liés à la complexité technique, à la coordination inter-équipes et à la maturité organisationnelle. Malgré ces avancées, un écart persiste entre la vision théorique des MSA et leur mise en œuvre réelle. Concernant la réutilisation, les travaux restent morcelés. Les études abordent soit l'extraction de microservices, soit la modernisation de systèmes hérités, soit des cadres conceptuels visant à structurer la réutilisation. Toutes convergent vers un même constat : la

réutilisation est souvent présentée comme un avantage des microservices, mais elle demeure difficile à atteindre en pratique, faute de standardisation, de gouvernance et de processus clairement établis. Dans l'ensemble, la revue met en évidence une lacune importante : peu de recherches analysent la réutilisation des microservices dans un contexte DevOps réel, en intégrant simultanément les dimensions techniques, organisationnelles et culturelles. C'est précisément ce vide que la présente étude cherche à combler.

CHAPITRE 2
METHODOLOGIE

Notre recherche adopte une démarche exploratoire qui s'appuie sur les recommandations de (Linåker et al., 2015). L'objectif est de comprendre comment la réutilisation des composants est réellement pratiquée sur le terrain dans le contexte MSA et DevOps. Pour réaliser cette étude, nous avons opté pour l'approche par enquête. Cette méthodologie nous propose des lignes directrices couvrant chaque étape qui doit être réalisée tout au long de notre étude telles que : la conception des questionnaires, l'identification des participants cibles, la diffusion du formulaire et le suivi des réponses pour la réussite de l'enquête, ainsi que la méthode d'analyse correspondante. Mais avant, nous allons décrire la stratégie de recherche que nous avons menée.

Conformément aux recommandations de (Linåker et al., 2015), l'approche par enquête a été retenue car elle permet de collecter des données directement auprès de praticiens, dans leur contexte réel de travail, et d'obtenir une vision fidèle des pratiques de réutilisation en environnement microservices. Le workflow méthodologique adopte dans cette étude est illustre à la figure 1. Il présente les différentes étapes suivies, depuis la conception du questionnaire jusqu'à l'analyse des données, conformément aux recommandation de (Linåker et al., 2015).

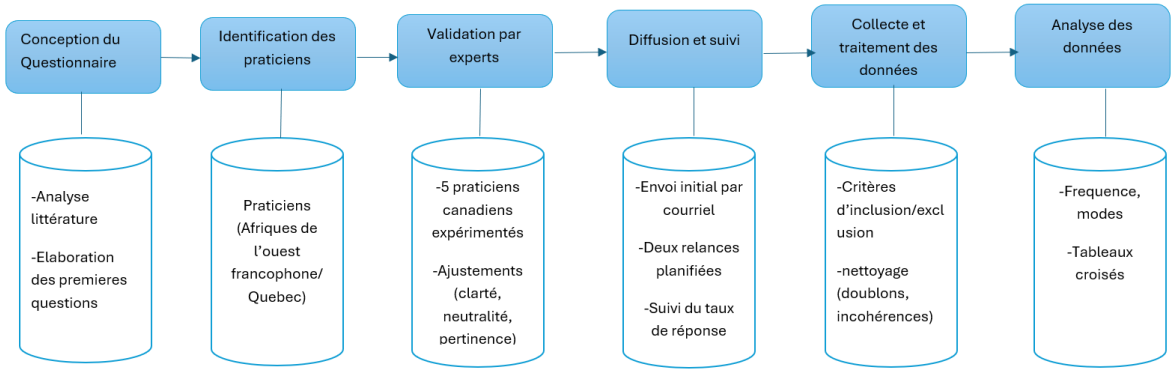


Figure 1: Workflow méthodologique de l'étude, adapté des recommandations de Lin et de Mell

Ce diagramme met en évidence la progression logique de la démarche, en montrant comment chaque étape s'appuie sur la précédente pour assurer la validité et la rigueur de l'enquête.

1. CONCEPTION ET STRUCTURATION DU QUESTIONNAIRE

Une revue de littérature a d'abord été réalisée. Celle-ci nous a permis d'identifier les modèles de qualité pertinents à partir desquels nous avons dérivé des premières questions : le modèle de maturité DevOps proposé par (de Feijter et al., 2018) et le modèle de réutilisation logicielle présenté par (W. Frakes & Terry, 1996). Dans notre étude, la réutilisation est considérée non seulement comme une pratique technique, mais également comme un attribut de qualité interne essentiel pour évaluer la valeur des logiciels fondés sur une architecture à base de microservices (MSA). Afin d'évaluer cet attribut dans un contexte réel, nous avons intégré dans notre questionnaire des items permettant de mesurer l'existence de pratiques de réutilisation à travers des indicateurs observables et mesurables. Le tableau 2 présente ces indicateurs et illustre la manière dont ils ont été intégrés dans le questionnaire.

A partir de cette première base, nous avons sélectionné les questions les plus pertinentes et en avons ajouté d'autres en fonction de nos connaissances du domaine. Ce travail a conduit à l'élaboration d'une première version du formulaire. Conformément aux recommandations de (Linåker et al., 2015), cette version préliminaire a été soumise et évaluée par des experts afin de s'assurer la clarté des questions et leur adéquation avec nos intentions de recherche. Les experts ont été invités à commenter la formulation, la compréhension, la neutralité et la pertinence des items. Leurs retours ont permis d'ajuster plusieurs questions afin d'éviter les ambiguïtés et de renforcer la validité du contenu.

Une fois corrigé et validé, le formulaire a été diffusé par courriel via de l'outil Google Forms auprès des participants préalablement identifiés. Le questionnaire a été rédigé en français, dans un langage simple et accessible, afin d'assurer une bonne compréhension

par l'ensemble des participants. Une version anglaise était également disponible au besoin.

Le questionnaire comporte un total de 20 questions, dont 19 sont des questions fermées et d'une question ouverte pour permettre aux répondants d'exprimer librement leurs opinions et expériences. Il s'ouvre par une courte introduction présentant les objectifs de l'étude et contextualisant la participation. Le questionnaire est structuré en trois sections :

- S1 Microservices (questions Q4 à Q8) : évaluation du niveau d'adoption des architectures à base de microservices ;
- S2 Pratiques de réutilisation dans les microservices et lien avec DevOps (questions Q9 à Q14) ;
- S3 Facteurs facilitateurs et obstacles à la réutilisation (questions Q15 à Q19).

Le questionnaire a été validé par un groupe de cinq praticiens chevronnés canadiens du génie logiciel (plus de 10 ans d'expérience). Ils ont été sollicités pour évaluer la clarté et la pertinence des questions. Leurs retours ont permis d'ajuster la formulation de certains items afin d'en améliorer la compréhension.

Pour évaluer les dimensions liées aux microservices et a DevOps dans un contexte réel, une partie du questionnaire a été spécifiquement structurée autour de ces deux approches. Le tableau 3 présente la correspondance entre les questions du formulaire et les concepts auxquels elles se rattachent.

Tableau 2: Correspondance entre les concepts théoriques de la réutilisation et des questions du formulaire de notre enquête

Concept théorique	**Dimension de la réutilisation**	**Questions correspondantes dans le questionnaire**
Intensité de la réutilisation	Fréquence de réutilisation	Au cours des deux dernières années combien de fois avez-vous réutilisé des microservices ?
Bénéfices attendus	Motivation de la réutilisation	Pourquoi réutilisez-vous ?
Réutilisation systématique/opportuniste	Type de réutilisation	Comment définissez-vous votre processus de réutilisation ?
Réutilisation interne/externe	Origine des microservices réutilisés	La majorité des microservices que vous avez réutilisés étaient créés par qui ?
Ownership et gouvernance	Propriétaire des microservices réutilisés	Qui était le propriétaire des microservices réutilisés ?
Qualité perçue de la réutilisation	Expérience de réutilisation	Comment qualifiez-vous votre expérience de réutilisation de micro-services comparée à celle de la réutilisation de services issus d'autres architectures (services légataire, monolithiques, etc.) ?

Absence de réutilisation	Duplication fonctionnelle	Au cours des deux dernières années, à quelle fréquence avez-vous fait des doublons ?
Facteurs favorables	Facilitateurs de la réutilisation	Quels sont les facilitateurs de la réutilisation dans ce contexte ?
Obstacle organisationnels et techniques	Barrière a la réutilisation	Quelle situation vous empêche de réutiliser ?
Processus décisionnel	Décision de réutilisation	De qui avez-vous besoin lors d'une décision de réutilisation ?

Tableau 3: Correspondance des questions du formulaire de notre enquête avec Microservices et DevOps

Question du questionnaire	**Approche en lien avec la question**
Avec quelle architecture travaillez-vous ?	Microservices
Participez-vous activement au développement des microservices ?	Microservices
Gérez-vous les microservices de bout en bout ?	Microservices
Parmi les défis suivants dans l'adoption et la pratique de l'architecture des microservices, auxquels faites-vous face ?	Microservices & DevOps

Est-ce que votre organisation vous donne les moyens de dépasser ces défis ?	DevOps
Au cours des deux dernières années combien de fois avez-vous réutilisé des microservices ?	Microservices
Pourquoi réutilisez-vous ?	Microservices
Comment définissez-vous votre processus de réutilisation ?	Microservices
La majorité des microservices que vous avez réutilisés étaient créés par qui ?	Microservices
Qui était le propriétaire des microservices réutilisés ?	DevOps
Comment qualifiez-vous votre expérience de réutilisation de micro-services comparée à celle de la réutilisation de services issus d'autres architectures (services légataire, monolithiques, etc.) ?	Microservices
Quels sont les facilitateurs de la réutilisation dans ce contexte ?	DevOps
Au cours des deux dernières années, à quelle fréquence avez-vous fait des doublons ?	DevOps
Quelle situation vous empêche de réutiliser ?	DevOps
Est-ce que c'est vous qui prenez la décision de réutilisation ?	DevOps
De qui avez-vous besoin lors d'une décision de réutilisation ?	DevOps

2. SELECTION DES PARTICIPANTS

La population cible de cette étude est constituée principalement de praticiens situés en Afrique de l'Ouest francophone et au Canada (Québec). Les participants de notre enquête ont été sélectionnés selon deux critères : (1) être localisés dans l'une de ces régions, et (2) être impliqués dans des projets utilisant une architecture à base de microservices (MSA), que cette implication soit actuelle ou prévue au sein de leur organisation. Sur cette base, la liste des participants cibles de notre enquête était préétablie.

3. LA DIFFUSION DU QUESTIONNAIRE ET SUIVI DE REPONSES

La diffusion du questionnaire a été réalisée directement par courriel auprès des participants ciblés. Un premier envoi a été effectué, suivi d'un premier rappel deux semaines plus tard afin d'encourager la participation. Une deuxième relance a ensuite été effectuée pour optimiser le taux de réponse. Au total, trois envois ont donc été réalisés : l'envoi initial et deux relances successives. Un suivi des réponses a été assuré tout au long de la période de diffusion afin de vérifier l'évolution du nombre de questionnaires complétés et de déterminer le moment opportun pour effectuer les rappels.

La période de collecte s'est déroulée du 20 janvier au 17 mars, ce qui a permis de laisser un délai suffisant entre chaque relance pour observer l'évolution naturelle du taux de réponse. Le suivi des réponses a été effectué via l'interface Google Forms, qui permettait de visualiser en temps réel l'évolution du nombre de questionnaires complétés. Les rappels ont été déclenchés lorsque le nombre de réponses cessait d'augmenter pendant plusieurs jours, ce qui justifie l'espacement des relances. Après chaque relance, une augmentation du nombre de réponses a été observée, confirmant la pertinence de cette stratégie de diffusion.

4. METHODE DE COLLECTE ET D'ANALYSE DE DONNEES

La collecte des données a été réalisée à l'aide de l'outil Google Forms, entre le 20 janvier et le 17 mars. Les participants ont été informés des objectifs de l'étude ainsi que du caractère confidentiel de leurs réponses. Les données ont été automatiquement réunies dans un document Google Sheets. Au total, dix-sept réponses ont été recueillies. Après application des critères d'inclusion (le répondant doit être impliqué dans un projet utilisant une architecture à base de microservices, que ce soit de manière actuelle ou planifiée) et d'exclusion (réponses incohérentes ou incomplètes), quatorze réponses valides ont été retenues pour l'analyse.

Le nettoyage des données a consisté à éliminer les réponses incomplètes, les doublons, ainsi que celles présentant des incohérences. Un tableau croisé dynamique a été utilisé pour repérer les contradictions ou les réponses hors sujet. Les données valides ont ensuite été analysées à l'aide de fréquences et de modes, les résultats étant synthétisés dans des tableaux croises réalisés avec Google Sheets. L'analyse a porté sur l'ensemble des questions, en se basant principalement sur les réponses des participants travaillant avec une architecture à base de microservices. Compte tenu de la taille réduite de l'échantillon (14), cette étude adopte une approche strictement exploratoire. Les analyses réalisées sont descriptives et ne visent pas à produire des généralisations statistiques, mais à dégager des tendances et des pistes de réflexion.

CHAPITRE 3

EXPLORATION DES PRATIQUES DE REUTILISATION DES MICROSERVICES DANS UN CONTEXTE DEVOPS

EXPLORATION DES PRATIQUES DE REUTILISATION DES MICROSERVICES DANS UN CONTEXTE DEVOPS

Elikee Wanda Messan[1], Morais Gabriel[1], and Mehdi Adda[1]

Université du Québec à Rimouski (UQAR), Canada
{ElikeeWanda.Messan, gabrielglauber.morais, mehdi_adda}@uqar.ca

RÉSUMÉ

Cette étude analyse les pratiques de réutilisation des microservices combinées à DevOps décrites dans la littérature et leur observabilité sur le terrain. Une en- quête par questionnaire a été menée auprès de dix-sept professionnels en Afrique de l'Ouest francophone et au Quebec (Canada). Les résultats montrent une adoption généralisée des microservices avec une réutilisation peu systématique, bien qu'elle soit émergente. Les facteurs techniques et organisationnels, tels que la documentation, l'automatisation des processus et la gouvernance influencent significativement cette pratique. Ces résultats mettent en évidence la nécessité de formaliser les pratiques de réutilisation dans des contextes microservices et DevOps.

Mots Clés : Architecture à base de microservices · DevOps · Réutilisation logicielle · qualité logicielle · attribut de qualité · recherche par sondage

1. Introduction

L'architecture à base de microservices (MSA) constitue une tendance croissante dans les organisations souhaitant moderniser leurs systèmes applicatifs existants [28]. En parallèle, DevOps émerge comme une approche culturelle et technique visant à rapprocher les équipes de développement et d'exploitation, dans le but de renforcer la collaboration, l'automatisation et la livraison continue des logiciels [17]. Ces deux paradigmes sont étroitement liés : les microservices sont souvent considérés comme un support architectural favorable aux pratiques DevOps, tandis que DevOps facilite l'implémentation efficace des architectures à base de microservices.
Dans ce contexte de transition vers des architectures plus modulaires et agiles, la réutilisation logicielle suscite un intérêt renouvelé. Traditionnellement considéré comme un levier permettant de réduire les coûts, d'améliorer la qualité logicielle et d'accélérer les cycles de développement [13], la réutilisation peine toutefois à s'imposer pleinement dans les environnements microservices. Cette difficulté persiste malgré son potentiel reconnu, en raison d'un ensemble de facteurs convergents d'ordre technique, organisationnel et procédural [21].

L'objectif de cette étude est d'examiner dans quelle mesure les pratiques de réutilisation des microservices décrites dans la littérature en lien avec la maturité DevOps se retrouvent également dans les pratiques observées sur le terrain. Pour atteindre cet objectifs, l'analyse est structurée autour de cinq question de recherche : (QR1) l'adoption et la gestion des microservices, (QR2) les objectifs associés à leur réutilisation, (QR3) les types de réutilisation observés, (QR4) les facteurs facilitant cette réutilisation et (QR5) les barrières qui la freinent. À travers cette recherche, notre étude contribue à une compréhension plus élargie des pratiques actuelles de réutilisation des microservices,

en comparant les recommandations issues de la littérature aux réalités empiriques. Pour réaliser cet objectif, une enquête par questionnaire structuré a été menée auprès de dix-sept praticiens, principalement issus de l'Afrique francophone et du Canada, afin de recueillir un aperçu concret des pratiques de réutilisation mises en œuvre dans des contextes professionnels variés.

Cette étude adopte une perspective exploratoire, en s'intéressant à l'état actuel des pratiques observées auprès de professionnels situés majoritairement en Afrique de l'Ouest francophone et au Québec (Canada). Bien que les résultats révèlent une réutilisation encore peu systématique, ils mettent en évidence des préoccupations récurrentes telles que la documentation, la maintenance et la coordination. Ces constats confirment que certains facteurs identifiés dans des travaux plus normatifs ou institutionnalisés sont également perceptibles dans des contextes moins formalisés. Cette comparaison suggère une convergence des défis tout en illustrant les écarts entre vision structurée et réalité du terrain.

Les travaux existants abordent diverses thématiques liées au MSA et à DevOps, notamment les écarts entre théorie et pratique des MSA, les défis et limitations associés aux outils DevOps, les facteurs de succès des projets DevOps, ainsi que la réutilisation des microservices dans le contexte DevOps. Ces études enrichissent la compréhension de l'adoption des microservices dans des contextes spécifiques, notamment sur les processus de migration ou l'intégration des pratiques DevOps, et confirment l'importance croissante de l'architecture à base de microservices en tant qu'architecture logicielle. Toutefois, la question de la réutilisation des microservices demeure relativement peu explorée, en particulier du point de vue des pratiques effectives sur le terrain. Les études existantes soulignent le potentiel des microservices à devenir des composants réutilisables, sous certaines conditions, et mettent en avant la contribution de cette réutilisation à la modularité et à l'évolutivité des systèmes. Notre étude s'inscrit dans cette perspective en analysant comment les microservices sont effectivement réutilisés selon des professionnels, surtout en Afrique.

L'article est structuré comme suit : la section 2 présente le contexte théorique, la section 3 décrit la méthodologie adoptée, la section 4 expose les résultats, la section 5 présente les travaux connexes, la section 6 discute les résultats et la section 7 conclut.

2. TRAVAUX CONNEXES

Les travaux portant sur les microservices et les pratiques DevOps ont évolué au fil des années, révélant progressivement les enjeux techniques et organisationnels liés à la réutilisation.

En 2019, [8] examinent la mise en œuvre des microservices dans un contexte industriel, en mettant l'accent sur l'extraction de services réutilisables à partir de systèmes existants. Leur étude montre que la variabilité constitue un élément central dans la structuration des microservices et que la simplicité des mécanismes permettant de gérer cette variabilité facilite leur réutilisation. Cette contribution met surtout en lumière les défis liés à la transition vers une architecture à base de microservices, mais elle reste centrée sur les aspects techniques de l'extraction plutôt que sur les pratiques organisationnelles qui influencent la réutilisation.

En 2023, [14] s'intéressent à la transformation d'un système d'examen en ligne en architecture basée sur les microservices, en mobilisant des règles de transformation centrées sur les fonctionnalités. Leur étude souligne l'importance de la performance, de la maintenabilité, de l'évolutivité et de la testabilité dans la modernisation des systèmes hérités. Bien qu'elle démontre que la réarchitecture peut renforcer la réutilisation, elle reste focalisée sur un cas particulier et ne traite pas des conditions organisationnelles nécessaires pour

soutenir cette réutilisation à grande échelle. Dans la même période, plusieurs travaux se penchent sur les défis rencontrés dans les environnements DevOps. À travers une enquête auprès de 21 professionnels, [26] identifient 23 préoccupations techniques et organisationnelles regroupées en quatre catégories : Cloud et CI/CD, Infrastructure as Code, conteneurs et architecture, et assurance qualité. Ces résultats montrent que les obstacles à l'adoption de DevOps dépassent largement les aspects techniques et incluent des enjeux de coordination, de culture et de gouvernance. Une revue de littérature menée par [2] confirme cette perspective en identifiant sept facteurs de succès des projets DevOps, répartis entre dimensions techniques, organisationnelles, sociales et culturelles. Ces travaux convergent vers l'idée que la réussite de DevOps repose sur un alignement entre outils, processus et culture, ce qui a des implications directes pour la réutilisation, même si celle-ci n'est pas étudiée explicitement. En parallèle, [20] explorent spécifiquement les défis de la réutilisation des microservices dans un contexte DevOps. Leur étude révèle que, malgré la perception des microservices comme naturellement réutilisables, leur réutilisation effective demeure limitée en raison de l'absence de processus structurés, de mécanismes de gouvernance et de standardisation. Cette contribution met en évidence un écart important entre les principes théoriques de modularité et la réalité opérationnelle des équipes. Plus récemment, en 2025, [1] proposent un cadre systématique visant à améliorer la réutilisation des microservices en répondant à des défis tels que la duplication de code, l'hétérogénéité technologique, la gestion des versions ou les décisions architecturales. Ce cadre offre une approche plus holistique, mais son application reste théorique et nécessite une validation empirique. Dans la même année, [3] analysent l'impact de l'adoption d'InnerSource et de pratiques CI/CD sur la réutilisation à grande échelle dans une étude de cas menée chez Ericsson. Ils observent une amélioration de la qualité et de la productivité, tout en soulignant les coûts initiaux liés à la conformité et à la standardisation. Cette étude montre que l'automatisation et la collaboration interne peuvent favoriser la réutilisation, mais qu'elles exigent des investissements organisationnels importants.

Enfin, [30] examinent l'écart entre la vision idéale des microservices et leur mise en œuvre réelle à travers des entretiens avec 20 professionnels en Chine. Leur étude met en évidence huit caractéristiques fondamentales de cette architecture, en distinguant les bénéfices perçus (tels que la flexibilité) des difficultés rencontrées, notamment la complexité de gestion et de surveillance. Ces résultats confirment que les microservices, bien qu'attrayants sur le plan conceptuel, posent des défis significatifs en pratique, notamment en matière de réutilisation.

Dans l'ensemble, la littérature révèle une progression des connaissances, mais aussi un manque d'études empiriques intégrant simultanément les dimensions microservices, DevOps et réutilisation. Les travaux existants abordent ces thèmes séparément ou partiellement, laissant un espace pour des recherches qui analysent la réutilisation dans des environnements DevOps réels et diversifiés : ce à quoi répond la présente étude.

3. CONTEXTE GÉNÉRALE

3.1. Qualité du logiciel

La qualité d'un logiciel se mesure par sa capacité à répondre aux besoins des clients tout en respectant les bonnes pratiques de développement. Plusieurs aspects peuvent être considérés pour en évaluer la qualité :

- La documentation et les tests : la documentation explique comment utiliser le logiciel, tandis que les tests permettent de vérifier son bon fonctionnement.
- Les processus de développement : ils regroupent les méthodes et

pratiques employées pour la création du logiciel.
- Les composants du logiciel : cela inclut le code, les fonctionnalités et l'architecture du système.

La qualité peut être abordée sous différents angles : la qualité interne (aspect lié au développement, les caractéristiques techniques) sur laquelle nous allons nous focaliser, et la qualité externe (respect des exigences, satisfaction des utilisateurs finaux). Pour évaluer pleinement la qualité logicielle, il convient de s'appuyer sur deux notions essentielles : les attributs de qualité et les modèles de qualité.

3.2. Attributs de qualité

La qualité logicielle peut être appréhendée à travers différents attributs qui influencent le comportement d'un système. Comme le souligne l'auteur, « C'est un ensemble de facteurs communs qui affectent un système lors de son exécution. D'un point de vue technique, les attributs de qualité déterminent d'importantes décisions en matière d'architecture et de conception » [27]. Mesurer la qualité d'un logiciel revient à vérifier s'il respecte les objectifs définis au départ et si le processus de développement répond aux besoins du client. Pour cela, différents attributs sont considérés, tels que :
-Réutilisation : capacité à réutiliser les modules développés pour de futurs projets, qui est l'attribut principal de notre étude.
-Fiabilité : mesure de la capacité d'un logiciel à exécuter ses fonctions prévues sans défaillance.
-Résilience : aptitude du logiciel à continuer de fonctionner malgré les défaillances de certains composants. La résilience est elle-même un indicateur de fiabilité.
Pour évaluer ces attributs, plusieurs modèles de qualité ont été proposés, parmi lesquels : le modèle de McCall, le modèle de Boehm, la norme ISO/IEC 25000, le modèle SQuaRE, les modèles de maturité DevOps, et le modèle de réutilisation. Dans le cadre de notre questionnaire, nous avons principalement construit les questions en nous appuyant sur le modèle de maturité DevOps proposé par [11] et le modèle de réutilisation de Frakes et Terry [12].

3.3. Réutilisation des logiciels

La réutilisation logicielle désigne le fait de construire de nouveaux systèmes à partir de composants existants plutôt que de repartir de zéro, comme le définissent les auteurs de [16]. Elle peut être soit systématique, lorsqu'elle repose sur des stratégies, des outils ou des infrastructures organisationnelles dédiées, soit opportuniste lorsqu'elle se fait de manière informelle et non planifiée.

3.4. Microservices

« Apparus comme une tendance ou un modèle issu d'une utilisation réelle, les microservices sont de petits services autonomes qui fonctionnent ensemble » [22]. Selon [4], chaque service peut disposer de sa propre installation de surveillance indépendante, appartenant à l'équipe d'opérations (Ops), ce qui permet un flux d'informations autonome entre performances et développement. [24] recommande la décomposition des systèmes en services simples, indépendants, qui communiquent via des API RESTful. À l'inverse, les systèmes monolithiques regroupent toutes les fonctionnalités dans une seule application, ce qui complique les modifications. Les microservices désignent à la fois une architecture et une approche de développement [31] : • En tant qu'architecture, ils se concentrent sur la structuration du système en services indépendants, favorisant la scalabilité, le déploiement indépendant, et la résilience. • En tant qu'approche de développement, ils privilégient l'autonomie des

équipes, les pratiques agiles, l'usage de technologies variées, ainsi que l'intégration et le déploiement continus. Ces caractéristiques permettent d'établir un lien étroit entre microservices et DevOps, qui partagent une vision centrée sur l'agilité et la collaboration. Ainsi, la migration vers une architecture à base de microservices peut favoriser l'adoption de pratiques DevOps.

3.5. DevOps

Avant l'émergence de DevOps, le développement logiciel suivait des modèles traditionnels, comme les approches en cascade ou en spirale, dans lesquelles les équipes de développement et d'opérations travaillaient de manière séparée.
Ces méthodes engendraient de longs cycles de livraison et de fréquentes erreurs en production. DevOps est né pour répondre à ces limites [15]. DevOps est une méthodologie de développement qui vise à rapprocher développement et opérations. Elle repose sur la communication, l'intégration continue, l'automatisation, la qualité, et la livraison rapide [11]. Elle favorise la collaboration entre équipes, l'automatisation des tâches, la surveillance des performances et l'amélioration continue. L'approche DevOps est issue de la méthode agile et s'inscrit dans un flux de travail continu [17]. Elle permet une évolution rapide et fréquente des produits, contrairement aux approches traditionnelles. DevOps et les microservices sont donc deux approches complémentaires qui visent une amélioration constante, une livraison continue, et une qualité optimale. Certains chercheurs ont suggéré une influence mutuelle entre ces deux pratiques. [5] affirme que « les microservices permettent une mise en œuvre efficace du DevOps en promouvant l'importance des petites équipes » et [9]de son côté soutient que l'architecture logicielle peut constituer un frein à l'implémentation de DevOps et à la livraison continue. En observant leurs caractéristiques communes « petites équipes, petits services, automatisation des tests et des déploiements, amélioration continue », il est légitime de considérer que les microservices et DevOps sont étroitement liés. Cette relation pourrait-elle se refléter dans certains attributs de qualité?

3.6. Attributs de qualité dans les microservices

De nombreuses études ont exploré les critères de qualité applicables aux architectures à base de microservices. Par exemple, [10] ont proposé un ensemble complet et applicable de critères d'évaluation. Dans une revue systématique (SLR), [18] identifient six attributs majeurs à partir de 72 études : l'évolution, la performance, la disponibilité, la surveillance, la sécurité et la testabilité. En complément, la nature même des microservices les rend propices à la réutilisation. [23] ont brièvement abordé la réutilisation dans le contexte des microservices. Cependant, cette réutilisation est-elle réellement observée dans les pratiques industrielles ? Si oui, et en considérant l'influence mutuelle entre MSA et DevOps, on peut se demander : quel impact cette relation exerce-t-elle sur la réutilisation et la fiabilité des logiciels, en particulier en termes de résilience applicative.

4. MÉTHODOLOGIE

4.1. Conception du Questionnaire

Le questionnaire de cette étude a été conçu en s'appuyant sur deux modèles existants : le modèle de maturité DevOps proposé par [20] et le modèle de réutilisation logicielle présenté par [12]. Ces modèles ont servi de base, mais ont été enrichis par des questions issues de nos propres connaissances, afin de couvrir les aspects spécifiques que nous souhaitions explorer, notamment la réutilisation dans les contextes microservices et DevOps. Dans le but d'explorer en profondeur pour une meilleur analyse et une crédibilité à notre étude , nous nous sommes inspirés du modèle existant de

vérification de la préparation DevOps (modèle de maturité DevOps proposé par [11] et du modèle de réutilisation proposé par [12] combinés à une revue de littérature évaluée par les pairs [10]. Le questionnaire a été rédigé en français, en utilisant un langage simple et accessible, afin d'assurer une bonne compréhension par tous les participants. Une version anglaise était également disponible en cas de besoin. Le questionnaire comporte un total de 20 questions, dont de 19 questions fermées, accompagnées d'une question ouverte pour permettre aux répondants d'exprimer librement leurs opinions et expériences. Il s'ouvre par une courte introduction présentant les objectifs de l'étude et contextualisant la participation. Le questionnaire est structuré en trois sections :

- T1 : Microservices (questions Q4 à Q8), visant à évaluer le niveau d'adoption des architectures à base de microservices ;
- T2 : Pratiques de réutilisation dans les microservices et lien avec DevOps (questions Q9 à Q14) ;
- T3 : Facteurs facilitateurs et obstacles à la réutilisation (questions Q15 à Q19).

Afin de mieux cerner le profil des participants, les trois premières questions du questionnaire (Q1 à Q3) portaient sur leur domaine d'activité, leur rôle au sein de l'équipe (ex. : développeur, opérateur, architecte logiciel, ingénieur qualité, gestionnaire), ainsi que leur nombre d'années d'expérience dans l'équipe. Enfin, la question Q20 est une question ouverte permettant aux participants de formuler librement leurs observations ou suggestions supplémentaires. Le questionnaire complet (Q1 à Q20) est fourni en Annexe (dans le mémoire) afin d'assurer la transparence méthodologique et de permettre la vérification de la correspondance entre les questions posées et les dimensions analysées. Les cinq questions de recherche (QR1 à QR5) ont été opérationnalisées sous forme de blocs thématiques dans le questionnaire. Chaque bloc correspond à une dimension spécifique de la réutilisation des microservices analysée dans les résultats. Cette structuration assure une cohérence entre les objectifs de recherche, la conception du questionnaire et l'organisation des résultats.

4.2. Praticiens cibles

La population cible de cette étude était constituée principalement de praticiens situés en Afrique de l'Ouest francophone (Togo, Sénégal) et au Canada (Québec). Les participants ont été sélectionnés selon deux critères :

1. Être localisés dans l'une de ces régions ;
2. Être impliqués dans des projets utilisant une architecture à base de microservices (MSA), que cette implication soit actuelle ou prévue au sein de leur organisation.

4.3. Validation du questionnaire

Le questionnaire a été validé en amont par un groupe de cinq praticiens canadiens du génie logiciel. Ils ont été sollicités pour évaluer la clarté et la pertinence des questions. Leurs retours ont permis d'ajuster la formulation de certains items pour une meilleure compréhension.

4.4. Diffusion et suivi

Une fois validé, le questionnaire a été diffusé directement auprès des participants ciblés.

Un premier envoi a été effectué, suivi d'un premier rappel deux semaines plus tard, puis d'une seconde relance afin d'optimiser le taux de réponse.

4.5. Collecte de données.

La collecte des données a été réalisée à l'aide de l'outil Google Forms, entre le 20 janvier et le 17 mars. Les participants ont été informés des objectifs de l'étude ainsi que du caractère confidentiel de leurs réponses. Les données ont été automatiquement réunies dans un document Google Sheets afin de simplifier leur traitement. Au total dix-sept réponses ont été recueillies. Après application des critères d'inclusion et d'exclusion (table1) : quatorze réponses valides ont été retenues pour l'analyse. Le nettoyage des données a consisté à éliminer les réponses incomplètes, les doublons et les incohérences. Les données valides recueillies ont ensuite été analysées par statistiques descriptives.

Table 1. Critères d'inclusion et d'exclusion des participants

Critères d'inclusion	• Le répondant doit être impliqué dans un projet utilisant une architecture à base de microservices (MSA), que ce soit de manière actuelle ou planifiée.
Critères d'exclusion	• Réponses incohérentes ou incomplètes.

5. RÉSULTATS

5.1. Données démographiques

Nous avons reçu un total de 17 réponses provenant de diverses organisations. La figure 1 illustre la répartition sectorielle des participants. La plus grande portion (28%) travaille dans les secteurs de la santé ou des services sociaux. 21% dans le commerce et la restauration. Ensuite, 14% des participants exercent dans le domaine de l'administration publique. Enfin, 7% sont issus de secteurs tels que la banque-finance, la gestion des stocks, l'éducation ou les technologies émergentes, ainsi que les technologies de l'information.

Une part importante des participants (35%) déclare avoir moins d'un an d'expérience dans leur équipe actuelle. Ensuite, 29% cumulent entre un et deux ans, ou entre deux et cinq ans d'expérience. Enfin, 5% déclarent une ancienneté comprise entre cinq et dix ans.

La majorité des répondants (52%) occupent un rôle d'opérateur. Les gestionnaires représentent 35% de l'échantillon, tandis que les développeurs en constituent 23%. (Les répartitions des personnes qui occupent plusieurs rôles ont été intégrés.)

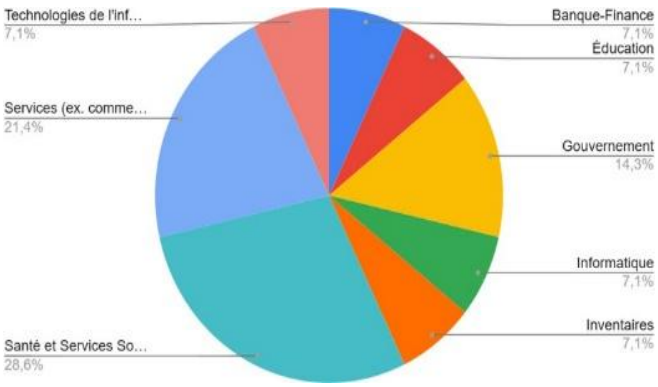


Figure 2. Secteurs de travail des participants

5.2. Conception des systèmes de microservices

Sur les 17 réponses reçues, seules 14 étaient complètes et pertinentes au regard des objectifs de l'étude. Les pourcentages présentés dans la section 4.1 sont donc calculés sur l'ensemble des répondants (n = 17), tandis que les analyses à partir de la section 4.2 se basent uniquement sur les réponses valides (n = 14).

La grande majorité des répondants (82%) déclarent travailler avec une architecture orientée microservices. À l'aide d'un tableau de croisement nous avons analysé les réponses concernant l'utilisation des MSA par domaine de travail, comme l'illustre la figure 2 ci-dessous. Pour la suite de notre analyse, un total de 14 réponses valides a été retenu.

QR1 : adoption et gestion des microservices

Parmi nos 82% répondants qui travaillent avec les MSA, nous avons 71% qui assurent la gestion des services de bout en bout, contre 11% qui ne participent pas. Concernant les défis rencontrés, les plus fréquemment cités sont : la décomposition appropriée des services (35%), la culture organisationnelle (35%). D'autres défis incluent l'automatisation du déploiement et la cohérence des données (14%), la surveillance, le débogage, la sécurité et la gestion des dépendances (7% chacun). Face à ces difficultés, 50% des participants estiment

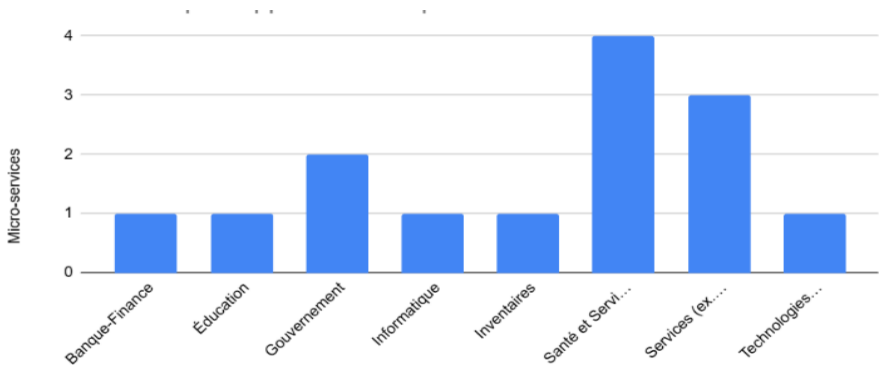


Figure 3. Microservices par secteur

que leur organisation met à disposition des moyens concrets pour y répondre, tandis que

35% évoquent un soutien partiel et 14% déclarent ne recevoir aucun appui.

QR2 : les objectifs de la réutilisation des MSA

La réutilisation des logiciels est l'utilisation des logiciels existants ou de connaissances logicielles pour construire de nouveaux logiciels [13]. Par conséquent, réutiliser un microservice revient à réemployer des microservices existants sans avoir à les redévelopper dans d'autres projets ou contextes. Concernant la fréquence de réutilisation, 50% des répondants indiquent avoir réutilisé des microservices moins de trois fois au cours des deux dernières années. Ensuite, 21% les ont réutilisés entre trois et six fois, et 14% plus de neuf fois. Les principales motivations évoquées sont l'amélioration de la stabilité (42%) et l'accélération du développement et des livraisons (42%). D'autres raisons, moins fréquentes, incluent la réduction des coûts (7%) et l'évitement des coûts de maintenance (7%).

QR3 : types de réutilisation utilisés

En ce qui concerne la nature de la réutilisation, 50% la qualifient d'opportuniste, tandis que 50% la considèrent comme systématique. Du point de vue de la provenance des microservices réutilisés : on a 57% qui ont été créés par un membre de leur propre équipe, 21% ont été développés par une équipe externe et 21% qui ont été développés par les répondants eux-mêmes. À l'échelle organisationnelle, les composants réutilisés proviennent 50% de leur propre équipe, 14% de leur département, 21% d'un individu, 7% d'un organisme externe et 7% de l'auto-réutilisation. Une majorité (78%) estime que la réutilisation de microservices est plus facile que celle de composants issus d'architectures monolithiques ou héritées. À l'inverse, 14% ne perçoivent aucune différence et 7% considèrent la réutilisation comme plus difficile.

QR4 : les facilitateurs à la réutilisation des MSA

Les facteurs facilitant la réutilisation sont : la disponibilité d'une documentation adéquate (28%), la régularité des évolutions et de la maintenance (7%), le fait que les composants aient été développés par leur propre équipe (49%), ainsi que d'autres facteurs (7% chacun) tels que : un développement personnel, le respect de normes et régulations, la compatibilité technologique, la présence d'un propriétaire identifié, ou l'indépendance des composants. Concernant les doublons, 57% des répondants indiquent avoir créé des microservices dupliqués moins de trois fois, 21% entre trois et six fois, et 21% n'en ont jamais créés.

QR5 : les barrières à la réutilisation des MSA

Les principales barrières à la réutilisation sont : le manque de documentation (50%), l'incompatibilité technologique (14%), les contraintes organisationnelles (14%), l'absence de maintenance ou d'évolution (7%), l'absence de responsable, un fort couplage ou la dépendance entre composants (7%), les exigences réglementaires (7%). Enfin, 78% des participants ne prennent pas eux-mêmes la décision de réutilisation, contre 21% qui le font. Les décisions sont majoritairement prises sur la base : des ressources techniques internes (57%), des ressources techniques du pro- duit à réutiliser (21%), du propriétaire du produit en cours de développement (7%), ou du propriétaire du produit à réutiliser (14%).

6. DISCUSSION

Ces résultats apportent un éclairage empirique sur les défis et les conditions favorables à la réutilisation dans différents contextes professionnels. Ils confirment que la réutilisation dans un environnement microservices dépend à la fois d'éléments architecturaux (granularité, découplage, modularité) et organisationnels (collaboration, rôles, documentation). La complémentarité entre MSA et DevOps, soulignée par certains auteurs (e.g. [4]), apparait également ici : les microservices fournissent une architecture flexible, tandis que DevOps apporte les processus

nécessaires à la livraison, l'intégration continue et la maintenance de services potentiellement réutilisables. [21]

Le manque de documentation adéquate, constitue un défi majeur freinant la réutilisation des MSA. Ce problème avait été souligné dans la littérature par [13] dans le contexte de la réutilisation logicielle. Outre la documentation, le manque de maintenance, l'absence de responsable désigné ou encore le couplage excessif entre services figurent parmi les obstacles relevés par les participants, en contradiction avec les principes des MSA.

Les résultats montrent une adoption importante des MSA (82,4% d'utilisation). Cependant, plusieurs obstacles limitent leur déploiement optimal, notamment la difficulté de décomposer les services de manière appropriée, les défis culturels organisationnels, et le manque d'automatisation. Ces constats rejoignent ceux de [4] et de [7], qui identifient ces facteurs comme critiques dans la transition vers une architecture à base de microservices.

Parmi les principaux facilitateurs de la réutilisation apparaissent la documentation claire, le développement au sein d'une même équipe, et la maintenance régulière. Ils soulignent également l'importance d'une gestion rigoureuse du cycle de vie des services pour rendre la réutilisation possible, comme le suggèrent [6]. Ces résultats montrent que la réutilisation n'est pas seulement un problème technique mais aussi organisationnel, le rôle de la documentation se révèle particulièrement déterminant.

La réutilisation des microservices demeure une pratique encore peu ancrée avec 50 % des répondants déclarant l'avoir expérimentée moins de trois fois en deux ans. Cette faible fréquence démontre une adoption encore émergente, plus particulièrement dans le contexte africain, où elle reste marginale. Ce constat rejoint les observations de [25], qui soulignent que la réutilisation est rarement systématique dans les environnements microservices en raison de l'absence de documentation, de standardisation et de gouvernance claire.

L'institutionnalisation de pratiques telles que la mise en place de bibliothèques partagées, de catalogues de services, ou des politiques de gouvernance contribuerait à renforcer la systématisation de la réutilisation, comme le préconisent [25]. Les études montrent que les défis organisationnels (tels que la coordination inter-équipes et la culture DevOps) constituent des freins récurrents, en complément des problématiques liées aux outils et à l'automatisation. Bien que nos résultats révèlent une réutilisation encore peu systématique, ils mettent en lumière des préoccupations similaires telle que la documentation, maintenance, coordination. Ils confirment que les facteurs identifiés dans des travaux plus normatifs ou institutionnalisés sont également perceptibles dans des contextes moins formalisés. Cette convergence souligne des défis communs, notamment la gestion de la connaissance, la gouvernance des services et la standardisation, tout en illustrant l'écart entre la vision théorique et la réalité du terrain.

6.1. Limites de l'étude

Cette étude comporte plusieurs limites. Tout d'abord, la taille restreinte de l'échantillon (n = 14) ne permet pas d'effectuer des vérifications statistiques inférentielles. Les résultats présentés demeurent strictement descriptifs et visent à dégager des tendances exploratoires. Ils ne prétendent pas à une validité statistique généralisable et doivent être interprétés avec prudence.

Les données recueillies mesurent la perception des professionnels et reposent sur des déclarations subjectives. Cette étude ne mesure pas les pratiques réelles, puisqu'aucune métrique logicielle objective n'a été extraite à partir de dépôts de code.

Le nombre restreint de réponses limite également la diversité des situations observées, ce qui réduit la représentativité des contextes organisationnels et techniques couverts par l'étude.

Ensuite, la majorité des participants sont issus d'une partie de l'Afrique (Afrique francophone). Ce contexte géographique et organisationnel spécifique peut influencer certaines pratiques, notamment en matière de ressources disponible, de maturité technologique ou de contraintes structurelles. Ainsi, les résultats ne reflètent pas nécessairement la réalité d'autres régions ou d'organisations opérant dans des environnements technologiques plus variés.

Par ailleurs, bien que le questionnaire permette de documenter la réutilisation des microservices et certains éléments contextuels, il ne mesure pas directement la relation entre DevOps et Microservices. L'étude ne permet donc pas d'établir un lien causal ou corrélationnel entre ces deux dimensions, ce qui constitue une limite importante au regard de la littérature existante.

Enfin, la méthode par questionnaire exclusif ne permet pas d'approfondir notre analyse. Cette méthode ne permet pas d'explorer les mécanismes internes, les processus décisionnels ou les dynamiques d'équipes qui influencent la réutilisation des microservices. Des méthodes qualitative (entretiens, études de cas, observations) auraient permis de compléter et d'enrichir l'interprétation des résultats.

6.2. Validité

La sélection des participants : nous avons envoyé les invitations à l'enquête par courriel sans connaître le niveau ou l'expertise des destinataires en microservices et DevOps. Pour pallier ce problème, nous avons inclus une section dans le formulaire pour préciser les caractéristiques des participants. Nous avons également utilisé nos contacts pour sélectionner des praticiens avec de différents profils. Questions fermées : presque toutes les questions de l'enquête sont fermées. Nous avons fourni un maximum d'options de réponse possibles en nous basant sur des sources existantes. Nous avons également inclus une zone de texte libre pour que les participants puissent ajouter d'autres réponses si nécessaire. Durée de l'enquête : notre questionnaire, composé de 21 questions, a été envoyé à environ vingt-trois participants et a duré trois semaines. Nous avons reçu dix-sept, dont quatorze étaient valides. Pour limiter les risques, nous avons choisi une période de l'année où les participants sont moins surchargés. Il faut environ 20 minutes pour compléter le sondage.
Validité externe : concerne la possibilité de généraliser les résultats de l'étude à d'autres contextes organisationnels. Echantillonnage et contexte : Notre enquête a été réalisée en collaboration avec une entreprise X[1] qui a accepté de participer à l'étude. Afin d'éviter un biais lié à un seul environnement, nous avons sélectionné des participants issus de l'Afrique francophone ouest (Togo, Sénégal) et du Québec (Canada). Cette approche visait à obtenir une représentation plus large de l'adoption des microservices et surtout des pratiques de réutilisation des microservices dans différents contextes et domaines d'activité.

[1] L'entreprise a demandé à rester anonyme afin d'éviter tout risque réputationnel lié à la divulgation des pratiques internes observées durant l'étude.

6.3. Perspectives

Des travaux futurs pourraient être réalisés en couvrant la partie de l'Afrique étudiée dans notre étude et d'autres partie afin d'avoir une vue un peu plus globalisante avec un nombre de répondants plus significatif que le nôtre. Une analyse comparative entre différents secteurs d'activité ou régions géographiques pourrait également permettre de mieux identifier les facteurs universels versus spécifiques à un contexte. L'étude du lien entre la maturité DevOps et la réutilisation systématique de services pourrait offrir des pistes intéressantes pour les organisations souhaitant institutionnaliser cette pratique. Enfin, d'autres méthodes de recherche, telles que des entrevues ou des approches qualitatives pourraient être réalisées en approfondissant l'exploration des processus interne de décision.

7. CONCLUSION

Cette étude visait à déterminer si les pratiques de réutilisation des microservices décrites dans la littérature, ainsi que leur articulation avec DevOps, se retrouvent dans les environnements industriels. À travers une enquête menée principalement auprès de professionnels situés en Afrique, nous avons obtenu un aperçu concret des pratiques en place, des défis rencontrés et des facteurs perçus comme influençant la réutilisation dans différents environnements organisationnels. Les résultats recueillis indiquent une adoption fréquente des microservices, avec une forte implication dans le développement et la gestion de bout en bout de ces architectures. Toutefois, la réutilisation de microservices est perçue comme peu systématique, bien qu'émergente. Les répondants associent cette situation a plusieurs facteurs techniques (tels que la qualité de la documentation, l'automatisation des processus ou le degré de couplage) et organisationnels (comme les rôles définis, la coordination entre équipes et les processus de décision). Les données suggèrent également que, selon les participants, la combinaison MSA/DevOps pourrait favoriser une réutilisation plus fiable, en renforçant la modularité, l'automatisation et l'isolation des défaillances. Ces perceptions soulignent la nécessité d'institutionnaliser davantage la réutilisation dans les environnements MSA/DevOps, notamment par la mise en place des bibliothèques de services partagées, une documentation normalisée, ainsi que des mécanismes de gouvernance adaptés. Ces résultats doivent toutefois être interprétés avec prudence en raison de la taille limitée de l'échantillon, ce qui inscrit cette étude dans une perspective exploratoire. Ces résultats ouvrent également la voie à des travaux futurs portant sur l'évaluation empirique de l'impact de la réutilisation sur la fiabilité et la résilience des systèmes distribués.

RÉFÉRENCES

1. Ait Said, M., Belouaddane, L., Mihi, S., Ezzati, A.: A systematic framework to enhance reusability in microservice architecture. Int. J. Comput. Digital Syst 16(1), 189–203 (2024)
2. Azad, N., Hyrynsalmi, S.: Devops critical success factors—a systematic literature review. Information and Software Technology 157, 107150 (2023)
3. Badampudi, D., Usman, M., Chen, X.: Large scale reuse of microservices us- ing ci/cd and innersource practices-a case study. Empirical Software Engineering 30(2), 41 (2025)
4. Balalaie, A., Heydarnoori, A., Jamshidi, P.: Microservices architecture enables de- vops: Migration to a cloud-native architecture. Ieee Software 33(3), 42–52 (2016)
5. Bass, L., Weber, I., Zhu, L.: DevOps: A software architect's perspective. Addison- Wesley Professional (2015)
6. Bhardwaj, C., Vyas, G., Sharma, A., Pareek, A.: Developing a quality model to study the impact

of variability over the reusability and maintainability of software product line variability models. In: Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS 2020, pp. 73–76. Springer (2020)
7. Bogner, J., Fritzsch, J., Wagner, S., Zimmermann, A.: Microservices in industry: insights into technologies, characteristics, and software quality. In: 2019 IEEE in- ternational conference on software architecture companion (ICSA-C). pp. 187–195. IEEE (2019)
8. Carvalho, L., Garcia, A., Assunção, W.K., Bonifácio, R., Tizzei, L.P., Colanzi, T.E.: Extraction of configurable and reusable microservices from legacy systems: An exploratory study. In: Proceedings of the 23rd international systems and software product line conference-volume A. pp. 26–31 (2019)
9. Chen, L.: Microservices: architecting for continuous delivery and devops. In: 2018 IEEE International conference on software architecture (ICSA). pp. 39–397. IEEE (2018)
10. Di Francesco, P., Lago, P., Malavolta, I.: Architecting with microservices: A sys- tematic mapping study. Journal of Systems and Software 150, 77–97 (2019)
11. de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., Brinkkemper, S.: De- vops competences and maturity for software producing organizations. In: En- terprise, Business-Process and Information Systems Modeling: 19th International Conference, BPMDS 2018, 23rd International Conference, EMMSAD 2018, Held at CAiSE 2018, Tallinn, Estonia, June 11-12, 2018, Proceedings 19. pp. 244–259.Springer (2018)
12. Frakes, W., Terry, C.: Software reuse: metrics and models. ACM Computing Sur- veys (CSUR) 28(2), 415–435 (1996)
13. Frakes, W.B., Kang, K.: Software reuse research: Status and future. IEEE trans- actions on Software Engineering 31(7), 529–536 (2005)
14. Hamed, S.H.A.: Reusability of legacy software using microservices: An online exam system example. Journal of Al-Qadisiyah for computer science and mathematics 15(3), Page–35 (2023)
15. Jabbari, R., bin Ali, N., Petersen, K., Tanveer, B.: What is devops? a system- atic mapping study on definitions and practices. In: Proceedings of the scientific workshop proceedings of XP2016. pp. 1–11 (2016)
16. Krueger, C.W.: Software reuse. ACM Computing Surveys (CSUR) 24(2), 131–183 (1992)
17. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, P.: A survey of devops concepts and challenges. ACM Computing Surveys (CSUR) 52(6), 1–35 (2019)
18. Li, S., Zhang, H., Jia, Z., Zhong, C., Zhang, C., Shan, Z., Shen, J., Babar, M.A.: Understanding and addressing quality attributes of microservices architecture: A systematic literature review. Information and software technology 131, 106449 (2021)
19. Linaker, J., Sulaman, S.M., Höst, M., de Mello, R.M.: Guidelines for conducting surveys in software engineering v. 1.1. Lund University 50 (2015)
20. Morais, G.: Enabling systematic microservices reuse through devops. In: CAiSE (Doctoral Consortium). pp. 1–10 (2023)
21. Morais, G., Adda, M., Bork, D.: Breaking down barriers: Building sustainable mi- croservices architectures with model-driven engineering. In: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Lan- guages and Systems. pp. 528–532 (2024)
22. Newman, S.: Building microservices. " O'Reilly Media, Inc." (2021)
23. Osses, F., Márquez, G., Astudillo, H.: Exploration of academic and industrial evi- dence about architectural tactics and patterns in microservices. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceeed- ings. pp. 256–257 (2018)
24. Richards, M.: Microservices vs. service-oriented architecture. O'Reilly Media Se- bastopol (2015)
25. Taibi, D., Lenarduzzi, V., Pahl, C.: Continuous architecting with microservices and devops: A systematic mapping study. In: Cloud Computing and Services Science: 8th International Conference, CLOSER 2018, Funchal, Madeira, Portugal, March 19-21, 2018, Revised Selected Papers 8. pp. 126–151. Springer (2019)
26. Tanzil, M.H., Sarker, M., Uddin, G., Iqbal, A.: A mixed method study of devops challenges. Information and Software Technology 161, 107244 (2023)
27. Tapia, V.C., Gaona, C.M.: Research opportunities in microservices quality assess- ment: A systematic literature review. Journal of Advances in Information Technol- ogy 14(5) (2023)
28. Velepucha, V., Flores, P.: A survey on microservices architecture: Principles, pat- terns and

migration challenges. IEEE access 11, 88339–88358 (2023)
29. Waseem, M., Liang, P., Shahin, M., Di Salle, A., Márquez, G.: Design, monitoring, and testing of microservices systems: The practitioners' perspective. Journal of Systems and Software 182, 111061 (2021)
30. Zhou, X., Li, S., Cao, L., Zhang, H., Jia, Z., Zhong, C., Shan, Z., Babar, M.A.: Re- visiting the practices and pains of microservice architecture in reality: An industrial inquiry. Journal of Systems and Software 195, 111521 (2023)
31. Zimmermann, O.: Microservices tenets: Agile approach to service development and deployment. Computer Science-Research and Development 32, 301–310 (2017)

CONCLUSION GÉNÉRALE

Afin de clarifier les apports de cette étude, cette conclusion présente les réponses directes aux trois questions de recherche formulées dans ce mémoire. Concernant la première question de recherche (QR1), les résultats laissent entrevoir que la réutilisation des microservices est effectivement présente dans les organisations, mais de manière limitée et peu systématique. La moitié des répondants déclarent avoir réutilisé des microservices moins de trois fois en deux ans, et cette réutilisation se manifeste surtout par la consommation de services existants, la réutilisation de modèles d'API, de chaînes d'intégration CI/CD et de configurations DevOps. Les participants rapportent que les motivations principales sont l'amélioration de la stabilité et l'accélération du développement, ce qui laisse à croire que la réutilisation est bien observée, mais demeure largement opportuniste.

Pour la deuxième question (QR2), les répondants identifient comme facilitateurs la disponibilité d'une documentation adéquate, la compatibilité technologique, l'indépendance des composants, la présence d'un propriétaire identifié et surtout le fait que les microservices aient été développés par la même équipe des répondants. La majorité considère la réutilisation plus facile, ce qui, selon eux, pourrait être lié à la modularité MSA et de l'automatisation DevOps.

Enfin, pour la troisième question (QR3), les participants perçoivent comme principales barrières le manque de documentation, l'incompatibilité technologique, les contraintes organisationnelles, l'absence de maintenance, l'absence de responsable et le couplage entre composants. De plus, la décision de réutiliser n'est généralement pas prise par les développeurs eux-mêmes, ce qui limite l'adoption systématique. Ces perceptions pourraient contribuer à expliquer pourquoi la réutilisation reste partielle malgré une forte adoption des microservices.

Ces résultats montrent que l'interaction entre l'architecture à base de microservices et les pratiques DevOps pourrait contribuer à faciliter la réutilisation et, selon les participants, à améliorer la fiabilité perçue et la résilience applicative, grâce à l'isolation des défaillances (indépendance des composants) propre aux microservices et à la reproductibilité offerte par l'automatisation DevOps.

1. CONTRIBUTION A LA RECHERCHE

Cette recherche contribue au domaine académique en offrant un éclairage empirique sur les pratiques de réutilisation des microservices (MSA) associé au DevOps, un sujet encore peu documente dans la littérature. En confrontant les recommandations théoriques aux observations recueillies sur le terrain, l'étude suggère plusieurs facteurs techniques et organisationnels qui influencent la réutilisation, tels que la qualité de la documentation, le degré de couplage, l'automatisation ou encore la coordination inter-équipes. Elle contribue à éclairer la compréhension des dynamiques propres aux architectures à base de microservices et aux pratiques DevOps, tout en soulignant les écarts entre les modèles conceptuels et leur mise en œuvre réelle dans différents contextes organisationnels.

Sur le plan pratique, cette étude fournit aux professionnels un aperçu concret des pratiques actuelles et des défis rencontres dans deux contextes géographique distincts : l'Afrique de l'Ouest francophone (Togo, Sénégal) et le Québec (Canada). Les résultats recueillis montrent que, malgré une adoption fréquente des microservices parmi les répondants, la réutilisation demeure peu systématique et nécessite une formalisation accrue. Les participants soulignent l'importance de mettre en place des bibliothèques de services partagées, une documentation normalisée, des mécanismes de gouvernance adaptes et une intégration explicite de la réutilisation dans les processus internes de développement. Ces constats offrent des pistes d'amélioration directement mobilisables par les organisations souhaitant optimiser leurs pratiques et renforcer l'efficacité de leurs environnements microservices.

2. LIMITES DE L'ETUDE

Cette recherche présente plusieurs limites. Tout d'abord, le nombre restreint de répondants (14 professionnels) limite la portée généralisable des résultats et réduit la diversité des situations observées. De plus, la concentration géographique sur l'Afrique de l'Ouest (Togo, Senegal) et le Québec (Canada) peut introduire des biais contextuels liés aux pratiques locales, aux ressources disponibles ou aux structures organisationnelles.

Par ailleurs, le recours exclusif à un questionnaire ne permet pas d'explorer en profondeur les processus décisionnels internes, les dynamiques d'équipe ou les mécanismes organisationnels qui influencent la réutilisation. Des méthodes qualitatives (entretiens, observations) auraient permis d'enrichir l'analyse.

Enfin, bien que le questionnaire permette d'évaluer plusieurs pratiques associées à la maturité DevOps (notamment l'automatisation, le CI/CD et certains aspects architecturaux), il ne repose pas sur un outil psychométrique valide permettant de calculer un score global de maturité. Cette absence d'instrument standardisé limite la possibilité d'établir une relation statique formelle entre maturité DevOps et réutilisation. Les données recueillies permettent toutefois d'identifier des tendances qualitatives utiles pour comprendre les interactions entre ces deux dimensions.

3. PERSPECTIVE ET RECOMMANDATION

Malgré ces limites, cette étude ouvre plusieurs pistes pour la recherche et la pratique :

Cette étude ouvre plusieurs perspectives pour les recherches futures. Compte tenu des limites méthodologiques identifiées, il serait pertinent d'approfondir l'analyse de la relation entre la maturité DevOps et la réutilisation des microservices à l'aide d'outils de mesure plus robustes, notamment des instruments psychométriques validés permettant de calculer un score global de maturité. De futures recherches pourraient également explorer plus finement les dynamiques organisationnelles qui influencent la réutilisation, en mobilisant des approches qualitatives telles que les entretiens semi-directifs, les études de cas ou les groupes de discussion. Enfin, l'extension de l'étude à d'autres secteurs d'activité ou à d'autres régions permettrait d'élargir la portée des résultats et de comparer les pratiques de réutilisation dans des environnements technologiques et culturels variés.

Sur le plan pratique, les résultats de cette recherche suggèrent qu'il serait pertinent pour les organisations de formaliser davantage la réutilisation au sein de leurs processus DevOps. Les professionnels interrogés soulignent l'importance de mettre en place des bibliothèques de services partagées, d'adopter une documentation normalisée et de définir

des mécanismes de gouvernance adaptés afin de faciliter l'identification, la maintenance et la réutilisation des microservices. L'intégration explicite de la réutilisation dans les processus internes de développement (notamment dans les phases de conception, de revue et de déploiement) pourrait constituer un levier essentiel pour renforcer l'efficacité opérationnelle et réduire la duplication des efforts. Ces recommandations offrent ainsi des pistes concrètes pour les organisations souhaitant optimiser leurs pratiques et tirer pleinement parti des architectures à base de microservices.

Dans l'ensemble, cette étude semble apporter des éléments utiles à la théorie comme à la pratique, en identifiant les facteurs favorisants et les obstacles à la réutilisation dans les architectures modernes. Elle fournit une base initiale permettant d'orienter les futures investigations et améliorer les pratiques industrielles.

ANNEXE :
PRESENTATION DU QUESTIONNAIRE

Dans le cadre de notre étude, ce questionnaire, diffusé via Google Forms, a été conçu pour capturer l'état de la pratique en matière de réutilisation de microservices ainsi que le contexte organisationnel dans lequel cette réutilisation s'inscrit. Il comprend plusieurs sections sur le profil des répondants, les pratiques de réutilisation et des questions contextuelles permettant d'interpréter ces pratiques. Le formulaire complet est reproduit ci-après afin d'assurer la transparence du processus de collecte

Questionnaires d'enquête

Introduction :

Ce formulaire vise à capturer l'état de la pratique en réutilisation de microservices et le contexte dans lequel cette réutilisation intervient. Notre objectif est d'identifier :

1. Les pratiques de réutilisation de microservices et leur relation avec le DevOps.
2. Les facilitateurs de la réutilisation.
3. Les barrières à la réutilisation.

Le formulaire contient des questions fermées ainsi que d'un espace, pour préciser vos réponses et émettre des commentaires. Votre collaboration est une aide précieuse pour faire avancer la connaissance sur les pratiques existantes et identifier des problèmes et des points d'amélioration.

* Indique une question obligatoire

1. Adresse e-mail *

2. Dans quel domaine travaillez-vous ? *

Une seule réponse possible.

- [ ] Industriel manufacturière
- [ ] Éducation
- [ ] Santé et Services Sociaux
- [ ] Services (ex. commerce, restauration, services à la personne)
- [ ] Banque-Finance
- [ ] Gouvernement
- [ ] Assurance
- [ ] Autre : _____

3. Depuis combien d'années travaillez-vous dans votre équipe actuelle ? *

Une seule réponse possible.

- [ ] < 1 an
- [ ] 1 a 2 ans
- [ ] 2 a 5 ans
- [ ] 5 a 10 ans
- [ ] 10 a 15 ans
- [ ]
- [ ] Autre : _____ >15 ans

4. Votre rôle(s) dans l'équipe *

Plusieurs réponses possibles.

- [ ] Développeur
- [ ] Opérateur
- [ ] Architecte Logiciel
- [ ] Ingénieur Qualité
- [ ] Gestionnaire

5. Avec quelle architecture travaillez-vous ? *

Plusieurs réponses possibles.

- [ ] Micro-services
- [ ] Services légataires/hérités (legacy services)

6. Participez-vous activement au développement de microservices ? *

Une seule réponse possible.

- ◯ Oui
- ◯ Non

7. Gérez-vous vos microservices de bout-en-bout ? *

Une seule réponse possible.

- ◯ Oui
- ◯ Non

8. Parmi les défis suivants dans l'adoption et la pratique de l'architecture des microservices, auxquels faites-vous face *

Plusieurs réponses possibles.

- [ ] Décomposition appropriée des services
- [ ] Sécurité
- [ ] Culture organisationnelle
- [ ] Automatisation du déploiement et de la gestion
- [ ] Maintien de la cohérence des données
- [ ] Gestion des dépendances
- [ ] Surveillance et
- [ ] débogage Autre : _____

9. Est-ce que votre organisation vous donne les moyens de dépasser ces défis ? *

Une seule réponse possible.

- ◯ Oui
- ◯ Non
- ◯ Partiellement

10. Au cours des deux dernières années combien de fois avez-vous réutilisé des microservices ? *

Une seule réponse possible.

- ◯ 0
- ◯ < 3
- ◯ 3 - 6
- ◯ 6 - 9
- ◯ > 9
- ◯ Autre : _____

11. Pourquoi réutilisez-vous ? *

Plusieurs réponses possibles.

- [ ] Pour accélérer les processus de livraison ou de développement ?
- [ ] Pour améliorer la stabilité ?
- [ ] Pour réduire les coûts
- [ ] Pour éviter les coûts de
- [ ] maintenance Pour des raisons
- [ ] environnementales ?

Autre : _____

12. Comment définissez-vous votre processus de réutilisation ? *

Une seule réponse possible.

- ◯ Systématique
- ◯ Opportuniste

13. La majorité des microservices que vous avez réutilisés étaient créés par qui ? *

Plusieurs réponses possibles.

- [ ] Par vous-même ?
- [ ] Par quelqu'un d'autre dans votre équipe ?
- [ ] À l'extérieure de mon équipe

14. Qui était le propriétaire des microservices réutilisés ? *

Plusieurs réponses possibles.

- [ ] Moi
- [ ] Mon équipe
- [ ] Ma section / département
- [ ] Mon organisation
- [ ] Organisation externe
- [ ] Je ne sais
- [ ] pas Autre :

15. Comment qualifiez-vous votre expérience de réutilisation de micro-services comparée à celle de la réutilisation de services issus d'autres architectures (services légataire, monolithiques, etc.) ? *

Une seule réponse possible.

- ◯ Plus facile
- ◯ Pareil
- ◯ Plus difficile

16. Quels sont les facilitateurs de la réutilisation dans ce contexte ? *

Plusieurs réponses possibles.

- [ ] Les composants ont été développés par moi-même
- [ ] Les composants ont été développés par mon équipe
- [ ] Propriétaire bien identifié
- [ ] Documentation adéquate accessible
- [ ] Evolution et maintenance régulière ?
- [ ] Stratégie de réutilisation commune à l'ensemble de l'organisation ?
- [ ] Présence de normes et réglementation ?
- [ ] Indépendance et faible couplage des composants ?
- [ ] Compatibilité technologique ?
- [ ] Autre :

17. Au cours des deux dernières années, à quelle fréquence avez-vous fait des doublons ? *

Une seule réponse possible.

- ◯ 0
- ◯ < 3
- ◯ 3 - 6
- ◯ 6 - 9
- ◯ > 9
- ◯ Autre : _____

18. Quelle situation vous empêche de réutiliser ? *

Plusieurs réponses possibles.

- ☐ Manque de documentation ?
- ☐ Manque de propriétaire ?
- ☐ Manque dévolution et maintenance ?
- ☐ Incompatibilité technologique ?
- ☐ Normes et réglementation ?
- ☐ Contrainte organisationnelle ?
- ☐ Dépendance et forte couplage (ex : monolithique) ?
- ☐ Autre : _____

19. Est-ce que c'est vous qui prenez la décision de réutilisation ? *

Une seule réponse possible.

- ◯ Oui
- ◯ Non

20. De qui avez-vous besoin lors d'une décision de réutilisation ? *

Plusieurs réponses possibles.

- [ ] Ressource technique dans votre équipe (architecte, assurance qualité, analyste fonctionnel)
- [ ] Propriétaire de votre produit
- [ ] Propriétaire du produit à réutiliser
- [ ] Ressource technique du produit à réutiliser
- [ ] Autre : _____

21. Si besoin, complétez vos réponses en donnant des détails dans cet espace.

_____

_____

_____

_____

REFERENCES BIBLIOGRAPHIQUES

1. Azad, N., & Hyrynsalmi, S. (2023). DevOps critical success factors—A systematic literature review. *Information and Software Technology*, *157*, 107150. https://doi.org/10.1016/j.infsof.2023.107150

2. Badampudi, D., Usman, M., & Chen, X. (2024). Large scale reuse of microservices using CI/CD and InnerSource practices—A case study. *Empirical Software Engineering*, *30*(2), 41. https://doi.org/10.1007/s10664-024-10595-w

3. Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional.

4. Boehm, B. W. (1978). *Characteristics of Software Quality*. North-Holland Publishing Company.

5. Bogner, J., Boceck, T., Popp, M., Tschechlov, D., Wagner, S., & Zimmermann, A. (2019). Towards a Collaborative Repository for the Documentation of Service-Based Antipatterns and Bad Smells. *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 95–101. https://doi.org/10.1109/ICSA-C.2019.00025

6. Carvallo, J. P., & Franch, X. (2006). Extending the ISO/IEC 9126-1 quality model with non-technical factors for COTS components selection. *Proceedings of the 2006 International Workshop on Software Quality, WoSQ '06*, 9–14. https://doi.org/10.1145/1137702.1137706

7. Chen, L. (2018). *Microservices: Architecting for Continuous Delivery and DevOps*. https://doi.org/10.1109/ICSA.2018.00013

8. Cojocaru, M.-D., Uta, A., & Oprescu, A.-M. (2019). Attributes Assessing the Quality of Microservices Automatically Decomposed from Monolithic Applications. *2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)*, 84–93. https://doi.org/10.1109/ISPDC.2019.00021

9. de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., & Brinkkemper, S. (2018). DevOps Competences and Maturity for Software Producing Organizations. In J. Gulden, I. Reinhartz-Berger, R. Schmidt, S. Guerreiro, W. Guédria, & P. Bera (Eds.), *Enterprise, Business-Process and Information Systems Modeling* (pp. 244–259). Springer International Publishing. https://doi.org/10.1007/978-3-319-91704-7_16

10. Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*, *150*, 77–97. https://doi.org/10.1016/j.jss.2019.01.001

11. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, Today, and Tomorrow. In M. Mazzara & B. Meyer (Eds.), *Present and Ulterior Software*

Engineering (pp. 195–216). Springer International Publishing. https://doi.org/10.1007/978-3-319-67425-4_12

12. Estdale, J., & Georgiadou, E. (2018). Applying the ISO/IEC 25010 Quality Models to Software Product. In X. Larrucea, I. Santamaria, R. V. O'Connor, & R. Messnarz (Eds.), *Systems, Software and Services Process Improvement* (pp. 492–503). Springer International Publishing. https://doi.org/10.1007/978-3-319-97925-0_42

13. Frakes, W. B., & Kang, K. (2005). Software reuse research: Status and future. *IEEE Transactions on Software Engineering*, *31*(7), 529–536. https://doi.org/10.1109/TSE.2005.85

14. Frakes, W., & Terry, C. (1996). Software reuse: Metrics and models. *ACM Computing Surveys*, *28*(2), 415–435. https://doi.org/10.1145/234528.234531

15. Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education.

16. Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). What is DevOps? A Systematic Mapping Study on Definitions and Practices. *Proceedings of the Scientific Workshop Proceedings of XP2016, XP '16 Workshops*, 1–11. https://doi.org/10.1145/2962695.2962707

17. Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution.

18. Larrucea, X., Santamaria, I., Colomo-Palacios, R., & Ebert, C. (2018). Microservices. *IEEE Software*, *35*(3), 96–100. https://doi.org/10.1109/MS.2018.2141030

19. Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A Survey of DevOps Concepts and Challenges. *ACM Comput. Surv.*, *52*(6), 127:1-127:35. https://doi.org/10.1145/3359981

20. Li, S., Zhang, H., Jia, Z., Zhong, C., Zhang, C., Shan, Z., Shen, J., & Babar, M. A. (2021). Understanding and addressing quality attributes of microservices architecture: A Systematic literature review. *Information and Software Technology*, *131*, 106449. https://doi.org/10.1016/j.infsof.2020.106449

21. Linåker, J., Sulaman, S., Host, M., & de Mello, R. (2015). Guidelines for Conducting Surveys in Software Engineering.

22. Mäkitalo, N., Taivalsaari, A., Kiviluoto, A., Mikkonen, T., & Capilla, R. (2020). On opportunistic software reuse. *Computing*, *102*(11), 2385–2408. https://doi.org/10.1007/s00607-020-00833-6

23. Mardiana, S., Tjakraatmadja, J. H., & Aprianingsih, A. (2015). DeLone–McLean Information System Success Model Revisited: The Separation of Intention to Use-Use and the Integration of Technology Acceptance Models. *International Journal of Economics and Financial Issues*, *5*(1), 172–182.

24. McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality:* Defense Technical Information Center. https://doi.org/10.21236/ADA049014

25. Morais, G. (2023). *Enabling Systematic Microservices Reuse Through DevOps*.

26. Morais, G., Adda, M., & Bork, D. (2024). Breaking Down Barriers: Building Sustainable Microservices Architectures with Model-Driven Engineering. *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, MODELS Companion '24*, 528–532. https://doi.org/10.1145/3652620.3687799

27. Newman, S. (2015). *Building microservices: Designing fine-grained systems* (First Edition). O'Reilly Media.

28. Richards, M. (2015). *Microservices vs. Service-Oriented Architecture*.

29. Taibi, D., Lenarduzzi, V., & Pahl, C. (2019). Continuous Architecting with Microservices and DevOps: A Systematic Mapping Study. In V. M. Muñoz, D. Ferguson, M. Helfert, & C. Pahl (Eds.), *Cloud Computing and Services Science* (pp. 126–151). Springer International Publishing. https://doi.org/10.1007/978-3-030-29193-8_7

30. Tanzil, M. H., Sarker, M., Uddin, G., & Iqbal, A. (2023). A Mixed Method Study of DevOps Challenges. *Information and Software Technology*, *161*, 107244. https://doi.org/10.1016/j.infsof.2023.107244

31. Tapia, V., & Gaona, M. (2023). Research Opportunities in Microservices Quality Assessment: A Systematic Literature Review. *Journal of Advances in Information Technology*, *14*, 991–1002. https://doi.org/10.12720/jait.14.5.991-1002

32. Team, D. (2021, March 15). DevOps: Qu'est-ce que c'est ? Principe, avantages, formation. *Formation Data Science | DataScientest.com*. https://datascientest.com/devops

33. Waseem, M., Liang, P., & Shahin, M. (2020). A Systematic Mapping Study on Microservices Architecture in DevOps. *Journal of Systems and Software*, *170*, 110798. https://doi.org/10.1016/j.jss.2020.110798

34. Waseem, M., Liang, P., Shahin, M., Di Salle, A., & Márquez, G. (2021). Design, monitoring, and testing of microservices systems: The practitioners' perspective. *Journal of Systems and Software*, *182*, 111061. https://doi.org/10.1016/j.jss.2021.111061

35. Zhou, X., Li, S., Cao, L., Zhang, H., Jia, Z., Zhong, C., Shan, Z., & Babar, M. A. (2023). Revisiting the practices and pains of microservice architecture in reality: An industrial inquiry. *Journal of Systems and Software*, *195*, 111521. https://doi.org/10.1016/j.jss.2022.111521

36. Zimmermann, O. (2017). Microservices tenets: Agile approach to service development and deployment. *Computer Science - Research and Development*, *32*(3–4), 301–310. https://doi.org/10.1007/s00450-016-0337-0