

Proposition COVEA

Thierry Antoni, Mheria

Version 1.0, Lundi 05 Mai 2025

Sommaire

1. Contexte et Objectifs	1
2. Pŕim•tre de la Proposition	2
2.1. Ce qui est inclus	2
2.2. Ce qui est exclu	2
2.3. Hypoth•ses	2
3. Solution ProposŽe	4
3.1. Architecture Technique	4
3.2. Description des Composants (Mise ^ jour et renforcement)	4
3.3. Choix Technologiques	5
4. Plan de Mise en Őuvre	6
4.1. Phases du Projet	6
4.2. Planning Prŕvisionnel	6
5. Livrables	8
6. Gouvernance et MŽthodologie	9
6.1. Organisation projet	9
6.2. MŽthodologie de travail	9
7. Estimation BudgŽtaire	10
7.1. Estimation en charge (jours-homme)	10
7.2. Conditions tarifaires	10
8. Risques et Mesures dŐAttŽnuation	11
9. Conclusion	12

1. Contexte et Objectifs

Dans le cadre de sa démarche de renforcement du dispositif de contrôle interne, Covža a déployé un outil dédié à la gestion et au suivi des contrôles.

À ce jour, l'outil présente plusieurs limitations techniques et fonctionnelles : instabilité, lenteurs, ergonomie perfectible, couverture partielle des besoins métiers, etc.

De plus, la gestion de l'outil reste fortement centralisée chez l'éditeur, limitant ainsi la capacité du client à le faire évoluer de manière autonome.

Dans ce contexte, Mheria se propose d'intervenir en tant qu'intégrateur pour :

- ¥ Stabiliser la solution existante en corrigeant les dysfonctionnements techniques et en fiabilisant les données
- ¥ Améliorer les fonctionnalités en lien avec les processus métiers de contrôle interne
- ¥ Structurer un dispositif de gouvernance technique et fonctionnelle clair et efficient
- ¥ Accompagner les équipes internes dans la montée en compétences afin de favoriser une appropriation progressive de l'outil
- ¥ Réduire la dépendance à l'éditeur à terme, en renforçant la capacité du client à administrer, faire évoluer et piloter l'outil en interne.
- ¥ Gérer la relation avec l'éditeur

L'objectif global de cette mission est de garantir la fiabilité et la performance de l'outil, tout en assurant son appropriation durable par les équipes de Covža, dans une logique d'autonomisation et de maîtrise du dispositif.

2. Pŕim•tre de la Proposition

2.1. Ce qui est inclus

L'intervention de Mheria couvre les activitŕs suivantes :

- ¥ Analyse de l'existant : revue des fonctionnalitŕs, identification des anomalies, diagnostic de la performance et des causes d'instabilitŕ.
- ¥ Stabilisation technique de l'outil : correction des dysfonctionnements, amŕlioration des performances, nettoyage de donnŕes si nŕcessaire.
- ¥ Amŕliorations fonctionnelles (MEV) : mise en ŕuvre d'ŕvolutions mineures/modŕrŕes, en lien avec les besoins mŕtiers identifiŕs lors de la phase d'analyse.
- ¥ Dŕfinition et mise en place d'une gouvernance de l'outil (technique et fonctionnelle).
- ¥ Accompagnement ^ l'appropriation : documentation, transfert de compŕtences, formation des ŕquipes internes ^ l'administration et ^ l'ŕvolution de l'outil.
- ¥ Pilotage du projet : coordination, suivi de planning, gestion des risques, reporting rŕgulier.
- ¥ Prise en charge de la relation contractuelle ou commerciale avec l'ŕditeur.

2.2. Ce qui est exclu

Les ŕlŕments suivants sont en dehors du pŕim•tre de cette proposition (sauf demande spŕcifique validŕe ultŕrieurement) :

- ¥ Refonte compl•te de l'outil ou changement de solution.
- ¥ Dŕveloppements lourds ou spŕcifiques non identifiŕs dans la phase d'analyse.
- ¥ Maintenance applicative ^ long terme (hors pŕriode de garantie si applicable).
- ¥ Intŕgration avec des syst•mes tiers (SI mŕtier, ERP, etc.) non documentŕe ou non prŕvue au dŕmarrage.

2.3. Hypoth•ses

La prŕsente proposition repose sur les hypoth•ses suivantes :

L'outil cible dŕj^ en production, accessible, et la documentation technique ou fonctionnelle disponible est fournie d•s le dŕbut de la mission.

Les interlocuteurs clŕs (mŕtier, IT, ŕditeur) sont mobilisŕs pour les phases d'analyse, de validation et de tests.

Les droits d'acc•s nŕcessaires ^ l'analyse et ^ l'intervention technique seront accordŕs ^ Mheria.

L'ŕditeur actuel reste disponible pour rŕpondre aux questions techniques ou fournir les informations nŕcessaires ^ la comprŕhension du socle de l'outil.

L'environnement technique de l'outil (infrastructure, hébergement, sécurité) est stable ou sous la responsabilité d'un tiers clairement identifié.

3. Solution Proposée

3.1. Architecture Technique

L'architecture cible reste centrée sur une organisation modulaire et structurée en couches fonctionnelles. L'objectif est de rendre le système plus résilient, mieux supervisé, et plus facilement maintenable. Les composants principaux restent inchangés (interface utilisateur, couche métier, base de données, intégrations SI), mais seront renforcés par des pratiques de qualité logicielle, de traçabilité et de contrôles en amont des traitements.

3.2. Description des Composants (Mise à jour et renforcement)

Optimisation de paramétrage (revue de code VB et SQL)

Une revue technique des modules paramétrés en Visual Basic et SQL sera réalisée, visant à :

- ¥ Identifier les anomalies de performance (requêtes inefficaces, code redondant)
- ¥ Proposer des améliorations de lisibilité et de maintenance
- ¥ Optimiser les interactions avec la base de données

Mise en place de règles de cohérence (à la saisie et à l'import)

Intégration de règles de validation côté interface et base :

- ¥ Contrôles bloquants ou alertes à la saisie utilisateur
- ¥ Filtres et nettoyages automatiques à l'importation de fichiers
- ¥ Mise en place de retours d'erreur explicites en cas d'incohérences

Mise en place de versionning et de code review via GIT Covža

Industrialisation des développements et paramétrages par :

- ¥ L'utilisation du dépôt GIT Covža pour versionner tous les paramétrages (VB/SQL/scripts)
- ¥ La mise en place de procédures de merge & pull request revues par un référent technique
- ¥ Un meilleur suivi des modifications, facilitant les audits, le rollback et la reprise après incident

Revue des flux avec contrôles et plans de secours

Un audit des flux d'intégration sera effectué, débouchant sur :

- ¥ La mise en place de contrôles de format, de correspondance référentielle, de volumétrie avant ingestion
- ¥ L'enrichissement des journaux de traitement pour un diagnostic plus rapide en cas d'échec
- ¥ La définition de mécanismes de repli ou de reprise manuelle assistée, avec journalisation des rejets

Revue de l'organisation des livraisons de paramétrage avec l'éditeur

Dans un objectif de traçabilité et de meilleure gouvernance :

- ¥ Mise en place d'un processus structuré de validation et de documentation des livraisons (Recette, Préprod, Prod)
- ¥ Tenue d'un journal centralisé des interventions externes (éditeur)
- ¥ Validation systématique sur environnement intermédiaire avant déploiement en production

Prise en charge de la montée de version Solstice (y compris Walling)

- ¥ Accompagnement technique et fonctionnel dans le passage à la nouvelle version Solstice
- ¥ Intégration du Walling et validation des impacts
- ¥ Réalisation de campagnes de tests de non-régression (manuels mais pourront être automatisés selon le contexte)
- ¥ Mise à jour des environnements de test avant mise en production

Rédaction des Spécifications Techniques Détaillées (STD)

- ¥ Production de documents techniques formalisant le comportement attendu des traitements, règles de gestion, flux, écrans
- ¥ Format standardisé, validé avec les interlocuteurs Covisa
- ¥ Utilisation des STD comme base de revue, validation, et documentation des évolutions

3.3. Choix Technologiques

Les actions s'appuient sur des technologies et outils standards :

- ¥ Langages : Visual Basic (paramétrage), SQL
- ¥ Contrôle de version : GIT Covisa (interopérabilité avec la forge interne)
- ¥ Outils de revue : GIT Pull Requests / Merge Reviews
- ¥ Gestion des imports/flux : Scripts d'automatisation (PowerShell, SQL Server Agent, etc.)
- ¥ Supervision / logs : Fichiers de trace enrichis, supervision applicative (à définir selon l'existant)

4. Plan de Mise en Œuvre

4.1. Phases du Projet

1. Phase 1 Ø Analyse, cadrage et préparation
 - ! Audit de l'outil existant (code VB, requêtes SQL, paramètres)
 - ! Revue des flux existants et des règles de gestion
 - ! Identification des points de blocage et des pistes d'optimisation
 - ! Organisation du GIT Covža (branches, process de revue)
 - ! Définition de la stratégie de tests de non-régression
2. Phase 2 Ø Stabilisation et optimisation
 - ! Optimisation des paramètres (VB/SQL)
 - ! Mise en place des contrôles de cohérence (saisie/import)
 - ! Implémentation des flux de contrôle et gestion des rejets
 - ! Structuration du processus de versionning (GIT)
 - ! Rédaction des premières STD associées aux corrections
3. Phase 3 Ø Montage de version Solstice et intégration Walling
 - ! Préparation des environnements (copie, anonymisation si besoin)
 - ! Installation de la nouvelle version (en coordination avec l'éditeur)
 - ! Réalisation des tests de non-régression fonctionnels et techniques
 - ! Revue des impacts sur les flux, écrans, règles de gestion
4. Phase 4 Ø Structuration de la gouvernance et documentation
 - ! Formalisation du processus de livraison (Rec / Préprod / Prod)
 - ! Mise en place du journal de livraisons de l'éditeur
 - ! Finalisation des STD
 - ! Formation des équipes internes si prévu
5. Phase 5 Ø Mise en production et accompagnement
 - ! Accompagnement à la mise en production
 - ! Suivi des premiers jours d'exploitation
 - ! Clôture et recommandations pour la phase de transfert

4.2. Planning Prévisionnel

ftape	Période estimée	Livrable clé
Phase 1 Ø Audit & cadrage	S1 à S2	Backlog, plan de tests, organisation GIT

ftape	PŽriode estimŽe	Livrable clŽ
Phase 2 Ð Optimisation & stabilisation	S3 ^ S6	Code optimisŽ, contr™les intŽgrŽs, STD v1
Phase 3 Ð MontŽe de version	S7 ^ S8	Nouvelle version validŽe, PV de tests
Phase 4 Ð Gouvernance & documentation	S9	Journal de livraison, STD v2
Phase 5 Ð Mise en production & suivi	S10	Mise en production validŽe, rapport final

5. Livrables

Phase	Livrables associés
Phase 1 : Analyse, cadrage et préparation	<ul style="list-style-type: none"> * Rapport d'audit technique (VB/SQL, flux) * Plan de tests de non-régression * Arborescence initiale du dépôt GIT Covža * Backlog des chantiers à engager (par ordre de priorité)
Phase 2 : Stabilisation et optimisation	<ul style="list-style-type: none"> * Code paramétré optimisé et documenté * Contraintes de cohérence opérationnelles (formulaires, imports) * Documentation des règles de contrainte mises en place * Premières versions des Spécifications Techniques Détaillées (STD) * Procédures de versionning et revue de code en production
Phase 3 : Montage de version Solstice + Walling	<ul style="list-style-type: none"> * Nouvel environnement installé et opérationnel * PV de tests de non-régression (fonctionnels et techniques) * Rapport d'impact de la nouvelle version * Plan de remédiation si anomalies détectées
Phase 4 : Gouvernance et documentation	<ul style="list-style-type: none"> * Procédure de livraison validée (REC/PRD) * Journal des interventions de l'éditeur (template et 1er suivi) * STD finalisées et validées (version 2) * Kit de transfert interne (si reprise de la main par Covža)
Phase 5 : Mise en production et accompagnement	<ul style="list-style-type: none"> * Rapport de mise en production * PV de recette définitive * Rapport de clôture du projet incluant recommandations

6. Gouvernance et Méthodologie

6.1. Organisation projet

Rôle	Responsabilités
Chef de projet client	Pilotage global, validation des livrables, arbitrage fonctionnel
Chef de projet prestataire	Coordination des travaux, planification, communication avec le client
Responsable technique	Suivi qualité du code, revue des paramètres, intégration des bonnes pratiques GIT
Développeurs / Paramètres	Mise en œuvre des optimisations, contrôles, tests, documentation
MOA / Utilisateurs clés	Expression des besoins, participation aux tests et validations
Fournisseur (tiers)	Support à la mise à jour de version, interventions encadrées sur l'outil

6.2. Méthodologie de travail

L'approche proposée est itérative et collaborative :

- Méthodologie hybride : cycle en V structuré pour la mise à jour de version et les phases critiques, complété par des sprints courts (2 semaines) pour les optimisations, tests et documentation.
- Réunions hebdomadaires de suivi d'avancement avec les parties prenantes.
- Comité de pilotage à chaque fin de phase, avec validation des livrables et arbitrages si nécessaire.
- Outils de suivi : tableau de bord de tâches (Excel, Jira ou outil Coviva), documentation partagée, backlog priorisé.

7. Estimation Budgétaire

7.1. Estimation en charge (jours-homme)

Phase	Estimation (JH)
Phase 1 : Cadrage et audit	8 JH
Phase 2 : Optimisation et contrôles	20 JH
Phase 3 : Montage de version + tests	12 JH
Phase 4 : Gouvernance / documentation	10 JH
Phase 5 : Mise en prod / accompagnement	5 JH
Total estimé	55 JH

7.2. Conditions tarifaires

¥ Tarification journalière selon grille en vigueur ou contrat-cadre.

¥ Toute évolution significative du périmètre fera l'objet d'un avenant ou chiffrage complémentaire.

¥ Déplacements éventuels sur site à prévoir en sus si non couverts par le forfait.

8. Risques et Mesures d'Atténuation

Risque	Impact potentiel	Mesure d'atténuation
Complexité du code existant non documenté	Ralentissement de l'audit, risques de régression	Phase d'analyse approfondie + tests de non-régression systématiques
Intervention parallèle de l'éditeur non coordonnée	Effets de bord imprévus, perte de traçabilité	Journalisation obligatoire + revue conjointe des livraisons
Non-disponibilité de certains référents côté client	Blocage de validation, prise de décision retardée	Planification en avance + substitution possible via backup de référents
Changements tardifs de version logicielle par l'éditeur	Tests obsolètes, retards de mise en prod	Veille continue + coordination proactive avec l'éditeur
Accès restreint à certains environnements ou outils (GIT, préprod)	Limitation des capacités d'intervention ou de test	Accords d'accès anticipés, création d'environnements clones si nécessaire

9. Conclusion

Cette proposition vise à assurer une montée en maturité de l'outil de gestion du contrôle interne utilisé par Covža, en alliant :

- ¥ Stabilisation technique via des optimisations ciblées
- ¥ Mise en place de bonnes pratiques de développement et de gouvernance
- ¥ Intégration sécurisée de la nouvelle version logicielle et de ses composants (Walling)
- ¥ Structuration documentaire avec les STD et les livrables de traçabilité

Nous nous engageons sur une approche structurée, progressive et transparente, visant à transférer la maîtrise de l'outil au client à moyen terme.

NOTE

Nous restons à disposition pour ajuster cette proposition en fonction de vos priorités, contraintes internes ou évolutions à venir. Un échange de cadrage complémentaire peut être organisé pour finaliser ensemble le plan de mise en œuvre.