

<b>1. Gestion des versions.....</b>	<b>2</b>
<b>2. Suivi des vulnérabilités.....</b>	<b>4</b>
<b>3. Mise à jour du système.....</b>	<b>7</b>
<b>4. Système de management de la conformité.....</b>	<b>7</b>
4.1 Définition des rôles et responsabilités.....	7
4.2 Cycle de vie du logiciel.....	8
4.2.1 Conception.....	9
4.2.2 Développement.....	9
4.2.3 Tests.....	10
4.2.4 Intégration.....	10
4.2.5 Fabrication.....	11
4.2.6 Déploiement/distribution.....	11
4.2.7 Configuration.....	11
4.2.8 Installation.....	11
4.2.9 Support.....	11
4.2.10 Évaluation et amélioration des performances du SMC.....	12
4.2.11 Traitement des anomalies.....	12
4.3 Contrôles de conformité.....	14
4.3.1. Conception.....	14
4.3.2. Développement.....	15
4.3.3. Tests.....	16
4.3.4. Intégration.....	17
4.3.5. Fabrication.....	17
4.3.6. Déploiement/distribution.....	17
4.3.7. Configuration.....	17
4.3.8. Installation.....	17
4.3.9. Support.....	17
4.3.10. Evaluation de l'efficacité du SMC.....	18
4.3.11. Traitement des non-conformités.....	19
<b>5. Gestion documentaire.....</b>	<b>19</b>
<b>6. Périmètre du code source.....</b>	<b>20</b>
6.1 Périmètre fiscal.....	20
6.2 Structure du code.....	21
<b>7. Maîtrise des sous traitants.....</b>	<b>22</b>
<b>8. Communication avec les clients.....</b>	<b>23</b>
<b>9. Licences du logiciel.....</b>	<b>24</b>
<b>10. Historique des irrégularités corrigés dans le logiciel, identifiées par le SMC...25</b>	<b>25</b>

12 avril 2025 - V4.2.14X.....	25
12 avril 2025 - V4.2.14X.....	25
31/07/2025 - V4.3.14X.....	25
30/01/2026 - V4.3.151.....	26
<b>11. Cahier de tests.....</b>	<b>31</b>
<b>11. Conformité au Référentiel de Certification.....</b>	<b>33</b>

## 1. Gestion des versions

La gestion du code source de kash.click est assurée via un dépôt SVN privé (svn://.../wwwCaisseEnregistreuse). SVN - svnserve, version 1.13.0 (r1867053) sur une machine virtuelle dédiée sous Ubuntu 22.04 (chez Digital Ocean, Europe)

Le numéro de version est composé de trois nombres ( VMajeure.VMineure.Revision )

- **Versions majeures** : Tout changement impactant le périmètre fiscal génère une version majeure (exemple : passage de 1.x à 2.x).
- **Versions mineures** : Évolutions fonctionnelles n'impactant pas la conformité fiscale (exemple : passage de 1.0.1 à 1.1.0).
- **Révision** : Corrections de bugs, évolutions mineures (exemple : passage de 1.0.1 à 1.0.2).

Politiques:

- **Politique de déploiement** : Mise à jour contrôlée et validation interne obligatoire avant toute mise en production.
- **Registre des systèmes et versions distribués** : le registre est disponible directement dans l'application en page Aide, rubrique Ressources. Le registre des versions en production dans le cloud est le dépôt SVN en lui-même.
- **Politique de communication aux clients finaux de la disponibilité des nouvelles versions majeures** : les nouvelles versions majeures doivent être notifiées systématiquement par trois moyens : publication en page d'accueil de l'information pendant les trente jours qui suivent la mise à jour + publication en page Aide, Actualités (l'historique des actualités), et envoi d'une information par email à tous les utilisateurs

Voir également : dossier de conception générale ; 2.2.2 Processus de mise à jour logiciel et 1. Gestion des versions

## 2. Suivi des vulnérabilités

- **Veille sécurité** : Surveillance continue des vulnérabilités PHP, MySQL et Linux.
- **Correction** : Déploiement des correctifs critiques.
- **Gestion d'incidents** : Procédure de traitement et d'escalade.

### **Modalités d'identification des vulnérabilités techniques dans le cadre de l'activité de développement**

L'éditeur met en œuvre plusieurs moyens pour identifier les vulnérabilités techniques au cours du cycle de développement du logiciel :

#### **Veille de sécurité régulière**

- Suivi des alertes de sécurité publiées par les fournisseurs des composants utilisés (frameworks, bibliothèques, serveur, OS, etc.).
- Abonnement à des sources officielles telles que CVE, NVD et les bulletins de sécurité des éditeurs concernés.

#### **Analyse de code et tests internes**

- Revue de code systématique lors des phases de développement critique (authentification, enregistrement des ventes, gestion des données sensibles).
- Utilisation d'outils d'analyse statique et/ou dynamique du code pour détecter les vulnérabilités connues (injections, XSS, stockage non sécurisé, etc.).
- Réalisation de tests fonctionnels et de non-régression avant chaque mise en production.

#### **Gestion des dépendances logicielles**

- Suivi des mises à jour de sécurité des dépendances tierces.
- Remplacement rapide de toute bibliothèque présentant une vulnérabilité identifiée.

#### **Procédures correctives**

- Les vulnérabilités identifiées sont enregistrées, évaluées (niveau de criticité), puis corrigées dans les plus brefs délais.
- Les correctifs sont intégrés dans une nouvelle version logicielle, testée avant déploiement.

## Séparation des environnements

- Les environnements de développement, de test et de production sont distincts, évitant l'exposition de données sensibles pendant les phases de test.

Outil de détection de failles de sécurités

La méthode choisie par l'éditeur net-assembly dans l'objectif de mettre en place en environnement sécurisé, ainsi que la surveillance des principales failles pouvant impacter l'application (injection SQL, logs leaks, DDoS, XSS, ...) sera d'utiliser l'utilitaire Nikto disponible en open source sur <https://github.com/sullo/nikto>

Cet utilitaire permet d'identifier :

- Interesting File / Seen in logs
- Misconfiguration / Default File
- Information Disclosure
- Injection (XSS/Script/HTML)
- Remote File Retrieval - Inside Web Root
- Denial of Service
- Remote File Retrieval - Server Wide
- Command Execution / Remote Shell
- SQL Injection
- File Upload
- Authentication Bypass
- Software Identification
- Remote Source Inclusion

Il conviendra d'exécuter la commande

`perl nikto.pl -h https://kash.click.local`

Avant chaque mise à jour majeure, afin de s'assurer que les conditions de sécurisation sont bien respectées. (Note interne : **utiliser la machine munin-Monitoring, puis cd /home/nikto/program puis perl nikto.pl -h <https://kash.click>** )

Certains avertissements pouvant cependant être ignorés si ceux-ci n'ont pas d'impact sur la sécurisation de l'application (urls vierges ou non utilisées, etc).

Pour exemple, résultat actuel de l'exécution de nikto, listant de nombreuses recommandations prévues pour être implémentées:

```
- Nikto v2.5.0
-----
+ Target IP:          82.196.0.218
+ Target Hostname:    kash.click
+ Target Port:        443
-----
+ SSL Info:           Subject: /CN=kash.click
                    Ciphers: ECDHE-RSA-AES256-GCM-SHA384
                    Issuer: /C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com,
Inc./OU=http://certs.godaddy.com/repository/CN=Go Daddy Secure Certificate Authority - G2
+ Start Time:         2026-01-16 14:42:32 (GMT1)
-----
+ Server: Apache
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content
of the site in a different fashion to the MIME type. See:
https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: Server may leak inodes via ETags, header found with file /robots.txt, inode: 1186502, size:
137, mtime: Tue Jun 11 15:40:14 2024. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /: The Content-Encoding header is set to "deflate" which may mean that the server is vulnerable to the
BREACH attack. See: http://breachattack.com/
+ /: Suggested security header missing: x-content-type-options. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options
+ /: Suggested security header missing: content-security-policy. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP
+ /: Suggested security header missing: permissions-policy. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Permissions-Policy
+ /: Suggested security header missing: strict-transport-security. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: Suggested security header missing: referrer-policy. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy
+ /index: Uncommon header(s) 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force
file names. The following alternatives for 'index' were found: index.php. See:
http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: DEBUG HTTP verb may show server debugging information. See:
https://docs.microsoft.com/en-us/visualstudio/debugger/how-to-enable-debugging-for-aspnet-applications?vie
w=vs-2017
+ /news/news.mdb: Web Wiz Site News release v3.06 admin password database is available and unencrypted.
+ /index.php?option=search&searchword=<script>alert(document.cookie);</script>: Mambo Site Server 4.0
build 10 is vulnerable to Cross Site Scripting (XSS).
+ /index.php?dir=<script>alert('Vulnerable')</script>: Auto Directory Index 1.2.3 and prior are vulnerable
to XSS attacks. See: https://vulners.com/osvdb/OSVDB:2820
+ /index.php?file=Liens&op=""<script>alert('Vulnerable')</script>: Nuked-klan 1.3b is vulnerable to
Cross Site Scripting (XSS). See: OSVDB-50552
+ /index.php?action=storenew&username=<script>alert('Vulnerable')</script>: SunShop is vulnerable to Cross
Site Scripting (XSS) in the signup page.
+ /index.php?action=search&searchFor=""<script>alert('Vulnerable')</script>: MiniBB is vulnerable to
Cross Site Scripting (XSS). See: http://www.minibb.net
+ /index.php/content/search/?SectionID=3&SearchText=<script>alert(document.cookie)</script>: eZ publish v3
and prior allow Cross Site Scripting (XSS).
+
+ /index.php/content/advancedsearch/?SearchText=<script>alert(document.cookie)</script>&PhraseSearchText=<sc
ript>alert(document.cookie)</script>&SearchContentClassID=-1&SearchSectionID=-1&SearchDate=-1&SearchButton
=Search: eZ publish v3 and prior allow Cross Site Scripting (XSS).
+ /?mod=<script>alert(document.cookie)</script>&op=browse: Sage 1.0b3 is vulnerable to Cross Site
Scripting (XSS). See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1243
+ /sitemap.xml: This gives a nice listing of the site content.
+ /index.php?rep=<script>alert(document.cookie)</script>: GPhotos index.php rep Variable XSS. See:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2397
+ /index.php?err=3&email=""<script>alert(document.cookie)</script>: MySQL Eventum is vulnerable to XSS in
the email field. See: OSVDB-12606
+ /index.php?vo=""<script>alert(document.cookie)</script>: Ralusp Sympoll 1.5 is vulnerable to Cross
Site Scripting (XSS). See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1175
+ /news: This might be interesting.
```

### 3. Mise à jour du système

- **Méthode** : Mise à jour par déploiement contrôlé via SSH sécurisé.
- **Traçabilité** : Chaque mise à jour est enregistrée (numéro de version, description des changements, date de déploiement).
- **Information client** : Notification aux clients de la disponibilité de nouvelles versions majeures.

### 4. Système de management de la conformité

#### 4.1 Définition des rôles et responsabilités

L'organisme met en place un système de management de la conformité (SMC).

Le responsable du SMC devra être désigné dans ce présent document et mise à jour en cas de modification.

Il aura pour responsabilité de:

- a) s'assurer que le système de management de la conformité est conforme au chapitre III du référentiel de certification;
- b) analyser les exigences techniques définies au chapitre IV du référentiel de certification;
- c) les décliner en spécifications fonctionnelles, pouvant être mises en œuvre ;
- d) assurer ou organiser des sessions de formation/information pour les employés concernés afin de s'assurer qu'ils soient conscients des exigences de conformité qui les concernent ;
- e) définir des indicateurs de performance de la conformité ;
- f) contrôler et mesurer ces indicateurs ;
- g) analyser les résultats pour identifier si des actions correctives sont nécessaires ;
- h) identifier et gérer les risques liés à la conformité relatifs aux tierces parties telles que fournisseurs, agents, distributeurs, consultants et sous-traitants ;
- i) superviser les conditions d'externalisation, afin de s'assurer qu'elles tiennent compte des exigences de conformité définies dans le présent référentiel.
- j) assurer une veille légale et réglementaire.

L'adresse [conformite@net-assembly.com](mailto:conformite@net-assembly.com) doit être à destination du responsable de la conformité.

Le SMC sera responsable du contrôle des exigences applicables au système d'encaissement.

## 4.2 Cycle de vie du logiciel

Lorsqu'une nouvelle version de l'application est poussée sur le serveur SVN de gestion de fichiers, une procédure supplémentaire devra être appliquée dans l'hypothèse où la révision contient des modifications qui portent sur des fichiers qui appartiennent au périmètre fiscal. Dans un tel cas, le SMC (système de management de la conformité) devra effectuer un audit interne des modifications apportées à ces fichiers afin de s'assurer que celles-ci n'auront pas d'impact négatif sur la conformité du système d'encaissement. Les contrôles devront s'appuyer directement sur le référentiel de certification des systèmes d'encaissement mis à disposition par l'organisme de certification.

Cet audit interne donnera lieu à l'émission d'un rapport qui sera inclus dans le dossier de maintenance.

En cas de non conformité l'analyse de la cause et les actions prises afin de corriger la non-conformité seront enregistrées dans le document de maintenance, et la mise à jour ne sera déployée qu'après l'application d'un correctif, et un nouveau cycle de mise à jour avec audit interne.

Ces contrôles pourront s'appuyer sur des politiques, procédures ou processus documentés mais également des approbations ou revues de code, des plans et rapport de test.

Durant le contrôle, il sera nécessaire de vérifier la signature cryptographique du périmètre fiscal, la conformité des signatures numériques attachées aux transactions.

- Tout commit ou push ou merge sur la branche principale SVN devra donner lieu à l'établissement d'un contrôle de conformité de mise à jour si des fichiers concernés appartiennent au périmètre fiscal où que le périmètre fiscal est modifié (cf chapitre "Contrôles de conformité").
- Une fois des évolutions fonctionnelles envoyées sur le serveur SVN, sur la branche principale, le code source pourra être déployé sur des serveurs de test. Ceci permettra de procéder au processus de test de ces nouvelles évolutions sur un environnement isolé factice et accessible uniquement en interne.
- Une fois la phase de test réalisée avec succès, la procédure de déploiement sur les serveurs de production pourra débuter. Celle-ci consiste en l'exécution d'un script permettant le déploiement simultané du code source sur tous les serveurs. Dès l'exécution du script terminée, la liste des évolutions fonctionnelles sera publiée sur la page d'accueil de l'application dans un encart intitulé "News" (changelog + actualités), permettant d'informer les utilisateurs des modifications apportées. La documentation est incluse dans le système

de gestion de versions de fichiers (SVN), et est mise à jour en même temps que l'application (la documentation est intégrée dans l'application et son code source est commun).

Responsable SMC : Simon Cabotse ([conformite@net-assembly.com](mailto:conformite@net-assembly.com))

#### 4.2.1 Conception

La conception de nouvelles versions pour le système d'encaissement suit un processus strict systématique :

- évaluation des besoins, estimation de l'impact technique des nouvelles évolutions fonctionnelles, basé sur une liste de recommandations ou de suggestions provenant soit des utilisateurs finaux, soit de l'équipe interne.
- à partir de cette évaluation des besoins sélection des évolutions fonctionnelles les plus pertinentes pour le logiciel
- pour chaque évolution fonctionnelle une nouvelle **étude d'impact** plus détaillée sera effectuée sur l'impact de la modification sur la conformité du système d'encaissement (notamment si celle-ci peut impacter les fonctionnalités d'enregistrement des données de vente, l'archivage des données, les fonctionnalités de clôture)
- Si l'évolution présente un impact sur la conformité, l'évolution fonctionnelle du système d'encaissement devra être validé par le responsable du SMC en interne
- Une fois les évolutions fonctionnelles sélectionnées, analysées, éventuellement validées par le SMC, l'équipe chargée du développement pourra procéder à la réalisation sur sa propre branche du code source.
- Si des fichiers appartenant au périmètre fiscal sont modifiés ou que le périmètre fiscal est modifié, la documentation réglementaire devra être mise à jour, et incluse dans le Commit sur la branche principale

#### 4.2.2 Développement

Lors de la phase de développement il conviendra de surveiller si la modification en cours de développement vient impacter ou non des fichiers appartenant au périmètre fiscal.

Si aucun fichier du périmètre fiscal n'est impacté et qu'il n'y a aucune fonctionnalité pouvant impacter les fonctions comprises dans les fichiers du périmètre fiscal, la modification en cours de développement pourra être développée sans procédure complémentaire.

Si la modification en cours de développement impact des fonctionnalités comprises dans le périmètre fiscal, il sera alors obligatoire de faire intervenir le responsable du système de management de la conformité afin qu'il procède à un audit du code impacté.

Cet audit devra avoir pour objectif de s'assurer de la conformité du système de caisse au cours de la modification en cours de développement. Il devra reprendre point par point chacun des éléments qui constituent le référentiel de certification et vérifier que pour chacun de ces points la conformité n'est pas remise en question.

#### 4.2.3 Tests

De la même manière si une modification en cours de développement impacte les fonctionnalités comprises dans le périmètre fiscal il sera alors obligatoire de faire intervenir le responsable du système de management de la conformité afin qu'il procède à la séquence complète de test prévu pour la vérification de la conformité du système de caisse.

Ces tests doivent particulièrement s'attacher à vérifier les fonctionnalités qui sont potentiellement impactées par la mise à jour.

Si la procédure de test du logiciel permet de mettre en évidence une rupture de la conformité du système d'encaissement, cela provoquera l'annulation de la mise en production du code source pour un retour en phase de conception, puis en face de développement jusqu'à ce que les tests veillant à surveiller la continuité de la conformité puisse être exécuté avec succès.

#### 4.2.4 Intégration

Les tests d'intégration du logiciel dans notre cas particulier bénéficient d'une simplicité due à la conception même du système d'encaissement.

En effet ce système d'encaissement est constitué à la fois d'une version Android, d'une version iOS, d'une version Windows, une version 32 bits, une version pour un navigateur , mais chacune de ces versions est en réalité composé d'un afficheur web permettant d'afficher la web app de l'application en plein écran sans outils de navigation afin de masquer le fait qu'il s'agisse d'une page web.

Cette particularité permet de conserver un très fort dynamisme dans la mise à jour de notre application et de n'avoir qu'une seule version à tester lors des tests d'intégration.

Les tests d'intégration se feront donc en utilisant un navigateur Google Chrome, le moteur utilisé par les versions Android, Microsoft, iOS de notre application.

#### 4.2.5 Fabrication

Aucune fabrication n'est effectuée par nos soins, nous proposons uniquement des solutions logicielles que le client peut installer sur un matériel compatible de son choix, qu'il doit acquérir par ses propres moyens auprès d'une entreprise tierce de son choix.

#### 4.2.6 Déploiement/distribution

L'architecture logiciel adoptée permet de garantir qu'à chaque instant une seule et unique version soit en permanence active et que cette version soit systématiquement la dernière version mise en production.

Du fait de l'architecture distante (dans le cloud) du logiciel, celui-ci n'a pas de code source déployé chez le client et une seule et unique version est utilisée par l'intégralité de la clientèle.

#### 4.2.7 Configuration

Ce logiciel ne dispose pas d'option de configuration super administrateur.

En d'autres termes, toutes les options de configuration sont des options de configuration utilisateur. Ces options ne peuvent pas impacter la conformité du système d'encaissement et n'ont d'impact que sur des options de personnalisation liées à l'activité du client final (configuration du nom de l'entreprise, du catalogue d'articles, etc).

Du fait de son architecture distante et de son déploiement universel, aucune configuration n'est à effectuer par un super administrateur pour le compte des commerçants.

#### 4.2.8 Installation

De la même manière que pour ce qui est de la configuration, du fait de son architecture dans le Cloud, ce logiciel ne nécessite pas d'installation particulière.

L'installation est effectuée directement par le client et elle ne peut pas avoir d'impact sur la conformité du système d'encaissement.

#### 4.2.9 Support

Tout utilisateur ou intervenant souhaitant obtenir des informations ou des explications concernant le système de management de la conformité ou bien la conformité en elle-même du logiciel pour un contacter directement par email le responsable du SMC à l'adresse suivante : [conformite@net-assembly.com](mailto:conformite@net-assembly.com)

Les utilisateurs du logiciel de caisse peuvent également contacter le support, via une interface dédiée intégrée directement dans le logiciel en page Aide, rubrique "Nous contacter".

Le support technique ne dispose d'aucun accès aux données du client, et peut uniquement effectuer des réponses aux questions posées.

Afin de garantir la non altération des données de l'établissement par le support technique, il ne sera jamais autorisé l'accès direct à la base de données dans le but d'effectuer le support, mais seulement à une interface spécifique directement dans le logiciel, permettant la

consultation des données. Les signatures cryptographiques permettant alors de bien attester de l'inaltérabilité des données.

#### 4.2.10 Évaluation et amélioration des performances du SMC

L'organisme met en œuvre une surveillance du SMC, qui consiste en la collecte et l'analyse d'information dans le but d'évaluer et améliorer l'efficacité du SMC. Il s'agira aussi bien : du résultat des tests de conception, de développement, ou encore des vérifications des sceaux cryptographiques via l'utilisation de l'outil prévu à cet effet.

Cette surveillance comprend l'évaluation de l'efficacité :

- des contrôles définis par exemple par l'analyse des résultats de test par échantillonnage
- du traitement des non-conformités précédemment identifiés
- des actions mises en œuvre pour réduire les risques liés à la conformité des systèmes d'encaissement distribués
- des prestataires externes

L'organisme tire ainsi parti de la surveillance du système de SMC afin de déterminer, mettre en œuvre et enregistrer toute action jugée pertinente permettant l'amélioration du SMC et la réduction des risques de non-conformité.

Au cours d'une réunion annuelle avec prospective de l'année l'entreprise étudiera les différentes manières d'améliorer le SMC, d'évaluer son efficacité, et effectuera un bilan sur les différents événements ayant survécu au cours de l'année.

Cette réunion aura pour but une introspection afin de déterminer si tous les moyens nécessaires et suffisants pour garantir la conformité au référentiel ont bien été mise en place.

Il s'attachera tout particulièrement à examiner entre autres :

- la sous-traitance
- le support client
- les fonctionnalités du logiciel ayant été mise à jour pendant la période

#### 4.2.11 Traitement des anomalies

Il ne peut exister aucune dérogation aux exigences du référentiel de certification des systèmes d'encaissement. L'organisme doit s'assurer que le système d'encaissement est bien conforme afin d'éviter la distribution et l'utilisation de système non conforme. En cas de détection d'anomalie au moment des contrôles, l'organisme réagira de la manière suivante : analyser l'anomalie en identifiant ses causes, mettre en œuvre des actions permettant de corriger l'analyse ou d'empêcher l'utilisation du système concerné évaluer l'efficacité des actions mises en œuvre, mettre à jour les risques identifiés, mettre à jour le SMC si nécessaire.

L'organisme enregistre les informations concernant la nature de l'anomalie, son analyse et les actions mises en œuvre avec leurs résultats dans le document si présent.

### 4.3 Contrôles de conformité

Les contrôles de conformité sont effectués par le SMC, sous la responsabilité du responsable SMC.

#### 4.3.1. Conception

Le contrôle portant sur la conception du système d'encaissement suit un processus strict systématique, portant sur la branche principale en production du système de gestion de versions de fichiers (SVN) :

- estimation de l'impact technique des évolutions fonctionnelles récentes
- estimation de l'impact technique de fonctions, basé sur une liste de recommandations ou de suggestions provenant soit des utilisateurs finaux, soit de l'équipe interne.
- vérification conceptuelle de chacun des points du référentiel de certification
- pour chaque fonction impactant le périmètre fiscal, une nouvelle étude plus détaillée sera effectuée sur l'impact de la fonction sur la conformité du système d'encaissement
- les fonctionnalités sélectionnées sont analysées, validées par le SMC
- vérification de la documentation réglementaire (mise à jour, et incluse dans le Commit sur la branche principale)

#### 4.3.2. Développement

Le contrôle portant sur le développement du système d'encaissement suit un processus strict systématique, portant sur la branche principale en production du système de gestion de versions de fichiers (SVN) :

- estimation de l'impact technique des évolutions fonctionnelles récentes
- estimation de l'impact technique de fonctions, basé sur une liste de recommandations ou de suggestions provenant soit des utilisateurs finaux, soit de l'équipe interne.
- vérification du code source de chacun des points du référentiel de certification
- pour chaque fonction impactant le périmètre fiscal, une nouvelle étude plus détaillée sera effectuée sur l'impact de la fonction sur la conformité du système d'encaissement
- les fonctionnalités sélectionnées sont analysées, validées par le SMC
- vérification de la documentation réglementaire (mise à jour, et incluse dans le Commit sur la branche principale)

Dans le cas d'une détection de NC :

- analyse de la cause d'une NC et des actions prises suite à celle-ci ;
- conception itérative corrective ;
- spécifications fonctionnelles liées à la conformité (III.3 & III.7) ;
- plans de test ;
- processus & méthode de développement (méthode propre à l'organisme) ;
- rapports de test;
- preuve de non impact sur la conformité d'une modification des processus de conception et développement;

#### 4.3.3. Tests

Il est obligatoire de faire intervenir le responsable du système de management de la conformité afin qu'il procède à la séquence complète de test prévu pour la vérification de la conformité du système de caisse.

Ces tests doivent particulièrement s'attacher à vérifier les fonctionnalités liées au périmètre fiscal et comprennent :

- la vérification des signatures cryptographiques
- inaltérabilité des données
- les sauvegardes et le bon état du processus automatisé

Le test doit comporter le scénario suivant

- Création d'un nouveau compte boutique sur la plateforme
- Enregistrement de ventes
- Validation des commandes
- Impression de justificatif
- Enregistrement de paiements
- Répéter la procédure sur plusieurs enregistrements
- Modifier la date système pour la journée suivante
- Répéter l'enregistrement de commande d'article et de paiement
- En page rapport télécharger un rapport
- Télécharger le fichier d'archive fiscale
- Vérifier que les signatures cryptographiques sont bien exactes\*
- Altérer le fichier (rapport) par une opération manuelle en modifiant directement les octets du fichier téléchargé
- Vérifier que la chaîne de signatures cryptographiques n'est plus valide à l'aide de l'outil de vérification des signatures
- Modifier directement dans la base de données un champ d'une donnée de vente : commencez par la table "articles" puis répéter l'opération pour les tables commande, paiement, fermetureAuto, traçabilité.
- Télécharger à nouveau le rapport et constater que les signatures sont bien indiquées comme incorrectes
- Repositionner le champ à sa valeur d'origine systématiquement

\*Les tests devront s'appliquer, en suivant un scénario similaire à celui décrit ci-dessus, à vérifier les signatures des tables des commandes, des articles, des paiements, des fermetures et des opérations d'archivage.

Il conviendra également de vérifier l'exactitude de la signature cryptographique du périmètre fiscal.

#### **4.3.4. Intégration**

Non applicable (cf 4.2.)

#### **4.3.5. Fabrication**

Non applicable (cf 4.2.)

#### **4.3.6. Déploiement/distribution**

Non applicable (cf 4.2.)

#### **4.3.7. Configuration**

Non applicable (cf 4.2.)

#### **4.3.8. Installation**

Non applicable (cf 4.2.)

#### **4.3.9. Support**

CF (cf 4.2.)

#### 4.3.10. Evaluation de l'efficacité du SMC

L'organisme met en œuvre une surveillance du SMC, qui consiste en la collecte et l'analyse d'information dans le but d'évaluer et améliorer l'efficacité du SMC. Il s'agira aussi bien : du résultat des tests de conception, de développement, ou encore.

L'organisme tire ainsi parti de la surveillance du système de SMC afin de déterminer, mettre en œuvre et enregistrer toute action jugée pertinente permettant l'amélioration du SMC et la réduction des risques de non-conformité.

#### **Collecte et d'analyse d'informations**

Les mesures de collecte et d'analyse d'informations (provenant des contrôles établis) permettant d'évaluer et améliorer l'efficacité du SMC doivent comprendre :

- des contrôles définis par exemple par l'analyse des résultats de test par échantillonnage
- du traitement des non-conformités précédemment identifiés
- des actions mises en œuvre pour réduire les risques liés à la conformité des systèmes d'encaissement distribués
- des prestataires externes
- des vérifications des sceaux cryptographiques via l'utilisation de l'outil prévu à cet effet
- des vérifications des copies de sauvegarde de la base de donnée
- des vérifications du serveur de réplication MySQL (slave status)
- de la vérification des modifications des fichiers du périmètre fiscal via l'outil SVN
- de la vérification un à un des points du référentiel de certification
- des vérifications sur les journaux d'événements (au niveau système, applicatif tiers, applicatif interne : logs d'erreur, événements à vérifier, etc)
- des alertes système (envoyées en cas de dysfonctionnement matériel, ou de dysfonctionnement majeur, ou d'incohérence de donnée)
- des signalements utilisateurs
- remarques de l'administration fiscale

#### 4.3.11. Traitement des non-conformités

Il ne peut exister aucune dérogation aux exigences du référentiel de certification des systèmes d'encaissement. L'organisme doit s'assurer que le système d'encaissement est bien conforme afin d'éviter la distribution et l'utilisation de système non conforme. En cas de détection d'anomalie au moment des contrôles, l'organisme doit réagir de la manière suivante : analyser l'anomalie en identifiant ses causes, mettre en œuvre des actions permettant de corriger l'analyse ou d'empêcher l'utilisation du système concerné évaluer l'efficacité des actions mises en œuvre, mettre à jour les risques identifiés, mettre à jour le SMC si nécessaire.

L'organisme enregistre les informations concernant la nature de l'anomalie, son analyse et les actions mises en œuvre avec leurs résultats dans le document si présent.

## 5. Gestion documentaire

Toutes les modifications sont documentées et intégrées aux dossiers techniques existants.

## 6. Périmètre du code source

### 6.1 Périmètre fiscal

L'intégralité des fonction ayant un impact important sur les fonctionnalités et exigences énoncées dans le référentiel de certification sont réunies le périmètre fiscal (Fichier "includes/corecert.php")

Certains fichiers annexes n'ont pas pu être fusionnés dans le fichier corecert.php :

- \$certifiedFiles[] = "includes/PeachSign.php";
- \$certifiedFiles[] = "includes/coreCert.php";
- \$certifiedFiles[] = "includes/export/archive.php";
- \$certifiedFiles[] = "workers/forms/statistiques/export.php";
- \$certifiedFiles[] = "workers/updaters/factureEnvoyee.php";

Ce fichier contient entre autres: le numéro de version majeur, toutes les fonctions liées aux signatures : Fonction de création du sceau cryptographique ; Fonction de validation d'une commande ; Fonction de génération de la clé cryptographique associée à la boutique; Fonction de protection de la modification des commandes validées ; Les fonctions de signatures

Fichiers non inclus dans le périmètre fiscal, mais dont la consultation apporte des informations complémentaires sur le logiciel, et dont la modification doit déclencher une vigilance de la part du responsable SMC :

Fichier "includes/article\_add\_check.php"

Contient les fonctions de vérification des conditions de création d'une nouvelle commande, édition de commande

Fichier "includes/shouldStartNewOrder.php"

Contient les fonctions de vérification des conditions de modification d'une commande

Fichier "includes/textes.php"

Contient les informations de révision (le troisième élément qui constitue le numéro de version)

Fichier "includes/perfStat.php"

Contient les fonctions d'accès aux données (fonctions DAO MySQL, important vis à vis de la protection contre les failles d'injection SQL)

Fichier “workers/forms/statistiques/export.php”

Les fichiers sont stockés dans un logiciel de gestion de versions de fichier (SVN).

L'historique des modifications apportées à ces fichiers y est clairement consultable, avec les dates de ces modifications, et les commentaires associés décrivant la modification apportée.

L'intégralité du code source étant d'ailleurs disponible par ce biais.

## 6.2 Structure du code

Le code est organisé avec une architecture de répertoires très simple :

- includes : contient les fichiers inaccessibles depuis l'extérieur
- workers : fichiers accessibles depuis l'extérieur en appel direct (points d'accès)
- les autres dossiers contiennent des medias ou des librairies (css, img, fonts, etc)

Dans chaque dossier, la racine contient les fichiers php génériques utilisés par de multiples pages ou fonctionnalités, tandis que les fichiers propres à des pages ou fonctionnalités particulières sont regroupés dans des sous dossiers.

## 7. Maîtrise des sous traitants

Afin de limiter les risques sur la conformité du système d'encaissement, la sous-traitance des activités de conception, développement, test, intégration, fabrication, support ne seront pas possibles et seront uniquement effectuées en interne.

- conditions de sous-traitance,

La sous traitance ne pourra être réalisée que pour des fonctionnalités non critiques, toute fonctionnalité critique devant rester interne. Les seules sous-traitances faisant exception sont l'architecture matérielle d'hébergement, la gestion des certificats SSL pour le protocole HTTPS, la fabrication, distribution de matériel de caisse, impression et expédition postale de factures.

Pour ces éléments sous-traités, il conviendra de s'assurer que l'organisme dispose bien des certifications adéquates en fonction de la responsabilité qui lui est confiée.

- résultats des processus externalisés,

La vérification du résultat des processus externalisés est très simple grâce au fait que la liste des fonctionnalités externalisée est réduite à son strict minimum nécessaire.

Il conviendra tout d'abord de s'assurer de la sécurisation des serveurs liés chez Digital Ocean.

Pour se faire il est indispensable de modifier le mot de passe par défaut du super administrateur Unix des machines dont nous avons la location ( et il faut ensuite vérifier que les mises à jour du système ont bien été effectuées).

- identification des fournisseurs/sous-traitants critiques,

La seule sous-traitance pouvant être considérée comme critique étant l'hébergement sécurisé. Il sera donc indispensable de s'assurer que chaque élément de matériel sur lequel le code source sera déployé soit bien un environnement contrôlé uniquement par l'équipe interne au système d'encaissement, et qu'aucun accès administrateur ne pourra être confié à aucune personne externe au système d'encaissement.

La personne responsable de la conservation et de la sécurisation des clés SSH permettant l'accès au serveur de production et au serveur de test devra être une seule et unique personne, interne à la société. Celle-ci devra être désignée dans le présent document et mise à jour en cas de modification.

Les seules activités dont la sous-traitance pourra être confiée à une entreprise tierce sera :

- la création de certificats de sécurité SSL sécurisé par des tiers

Les certificats de sécurité sont gérés par GoDaddy

- la mise en place est là maintenant d'infrastructure d'hébergement sécurisé (Cloud)

L'hébergement est réalisé par OVH, Digital Ocean

- la fabrication, distribution de matériel de caisse

Matériel de caisse fabriqué par Starmicronics, Sunmi, Pax, SumUp, distribution via Amazon

- impression et expédition postale de factures

Réalisé par Esker

- analyse de risques sur la conformité de la sous-traitance,
- actions pertinentes pour la réduction du risque lié à la sous-traitance.

Responsable de la conservation et de la sécurisation des clés d'accès : Simon Cabotse

## 8. Communication avec les clients

Le système prévoit la transmission à tous les clients chez qui le système d'encaissement est installé de tous les documents nécessaires au bon fonctionnement de celui-ci, des procédures de support et de formation, les engagements de responsabilité vis-à-vis de la loi des finances pour 2016, une description du moyen d'accès aux données d'encaissement par l'administration fiscale ainsi que d'un manuel utilisateur à destination de l'administration fiscale décrivant le moyen d'accès aux données d'encaissement, une description du format présenté, et la manière de procéder à la vérification d'intégrité des données.

Les documents précités seront disponibles pour les équipes internes et pour les utilisateurs pendant 3 ans après la date de fin de distribution de chaque version majeure du système d'encaissement.

Ces documents sont accessibles directement dans l'application, en naviguant en page Aide, rubrique Conformité du logiciel.

## 9. Licences du logiciel

La licence acquise par l'utilisateur et mémorisée en base de données dans la table dénommée "boutiques" via le champ intitulé **formulePremium** et la durée de la licence restante pour l'utilisateur et quand elle est stockée dans le champ **licencePremium**

**Les licences du logiciel correspondent à des comptes utilisateurs, chaque utilisateur dispose d'une licence, même ceux n'ayant pas souscrit à une licence payante.**

Les comptes des utilisateurs sont stockés en base de données (dans la base principale), dans la table utilisateurs.

Ceci permet à l'éditeur de garder une liste exhaustive et à jour des licences actives distribuées.

Si un utilisateur demande la suppression de ses données conformément à informatique et libertés, son compte utilisateur est déplacé dans une autre table (utilisateurs\_suppr) qui n'est pas accessible dans l'application et qui ne peut être consultée qu'en cas de demande par les services fiscaux. Les autres données liées au compte ne sont pas supprimées, mais deviennent inaccessibles, car leur accès est sécurisé par le fait qu'une connexion à un compte utilisateur valide et accrédité est nécessaire pour pouvoir y accéder. Il sera possible de restaurer l'accès à ces données, uniquement à la demande des services fiscaux, ou du client lui-même.

## 9. Inventaire des clients

Du fait de son architecture dans le cloud, l'application dispose à chaque instant d'une liste exhaustive de ses clients dans une base de données. Il est à tout moment possible de lister l'intégralité des clients du logiciel et de connaître également la version majeure qu'ils utilisent étant donné que l'intégralité des clients ne peut utiliser que la version majeure courante du logiciel( toujours du fait de l'architecture dans le Cloud de l'application).

Pour pouvoir utiliser l'application l'utilisateur a besoin de se connecter à son compte, et c'est justement son compte qui est en base de donnée, accessible via une web app qui est l'unique point d'accès de l'application. Qu'il s'agisse des différentes versions sous Android sous iOS ou sous Windows Mobile de l'application, chacune de ces applications intègre en réalité un navigateur Chromium ou Edge plein écran qui affiche l'application Web sur l'appareil et **n'est donc qu'une interface**. Les mises à jour sur l'APK ou le package iOS n'entraînent pas de modification de la version majeure mais uniquement des modifications mineur de fonctionnalités. La version de l'APK Android ou du package iOS n'est pas visible en base de données mais communiquée lorsque l'utilisateur effectue une demande d'assistance.

Le déploiement des mises à jour est instantané et universel, les utilisateurs obtiennent tous simultanément, instantanément, la dernière version mise en production.

## 10. Historique des irrégularités corrigés dans le logiciel, identifiées par le SMC

Toujours indiquer le numéro de version.

### 12 avril 2025 - V4.2.14X

Il a été identifié une irrégularité dans la génération des rapports qui permettait de générer des rapports portant sur une plage de date supérieure à un an. Un patch correctif a été appliqué au logiciel et celui-ci ne permet désormais plus de télécharger des rapports sur une plage de date supérieure à un an.

### 12 avril 2025 - V4.2.14X

Il a été identifié une irrégularité dans la traçabilité des opérations concernant les opérations d'archivage. Le système de caisse enregistrait bien les documents d'archive générés en base de données mais il n'enregistrait pas l'identifiant de l'utilisateur ayant effectué la génération de l'archive (enregistrait l'identifiant de boutique). Le logiciel a été mis à jour et l'identifiant de l'utilisateur est désormais également enregistré dans la base de données, dans une table dédiés aux trace des opérations, pour permettre une traçabilité plus précise des opérations d'archivage

### 31/07/2025 - V4.3.14X

Il a été identifié une erreur dans le lien vers la documentation pour la génération du SHA256 du code source du périmètre fiscal de l'application. Ce lien pointait vers une version .docx du document alors qu'il eut fallu utiliser un lien vers le fichier au format PDF. Les liens ont été corrigés, et le logiciel utilise bien le contenu des fichiers de documentation pour générer le SHA256. Le numéro de version mineure est incrémenté. La version passe à 4.3.000

**30/01/2026 - V4.3.151**

Lors de l'audit initial de l'application, les non conformités suivantes ont été relevées :

**- La documentation de conception générale ne référence pas le protocole réseau permettant l'accès au système (HTTP/HTTPS), et ne référence pas les versions TLS autorisés ;**

Plan d'action : Ajout d'informations dans dossier de conception générale ; 2.1 Architecture logicielle

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- l'organisation n'a pas formalisé l'architecture détaillée au sein du dossier d'architecture technique avec la nature des flux entre les différents composants du système (serveurs front à Amsterdam et les bases de données qui se trouvent à Amsterdam et Londres) ;**

Plan d'action : Ajout d'informations dans dossier de conception générale ; 2.1 Architecture matérielle

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Les processus de configuration et déploiement ne sont pas décrit dans le dossier organisationnel ;**

Plan d'action : Ajout d'informations dans dossier de conception générale ; 2.1.1 Processus de mise à jour logiciel

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Dossier de maintenance : La documentation fait mentionne l'usage d'outils d'analyse statique et dynamique du code pour détecter les vulnérabilités connues, mais ne les référence pas explicitement dans le chapitre relatif à la gestion des vulnérabilités ;**

Plan d'action : Ajout d'informations dans dossier de maintenance ; 2. Suivi des vulnérabilités

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- le dossier de maintenance ne référence pas le processus de gestion des licences implémenté par le système.**

Plan d'action : Ajout d'informations dans dossier de maintenance ; 9. Licences du logiciel

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Dossier de maintenance : Le processus de mises à jour d'une version n'est pas décrit dans le dossier de maintenance.**

## Dossier de Maintenance

Plan d'action : Ajout d'informations dans dossier de conception générale ; 2.2.2 Processus de mise à jour logiciel et 1. Gestion des versions

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- le dossier de maintenance ne comporte pas la politique de versionning du code mentionnant la gestion des versions majeures/mineures au sens de la réglementation.**

Plan d'action : Ajout d'informations dans dossier de maintenance ; 1. Gestion des versions ajout d'un lien vers les sections correspondantes dans le dossier\_de\_conception\_générale dossier de conception générale ; 2.2.2 Processus de mise à jour logiciel et 1. Gestion des versions

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- le dossier de maintenance ne comporte pas de description de l'architecture du code source (organisation des différents fichiers de code, branches sur le gestionnaire de code) ;**

Plan d'action : Ajout d'informations dans dossier de maintenance ; 6.2 Structure du code

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- La documentation à destination des utilisateurs est entièrement en ligne, sans présence de versionning permettant de tracer les évolutions de la documentation.**

Plan d'action : Ajout d'informations dans dossier de conception générale ; 6. Documentation

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Exigence 3 : les notes de frais ne fait pas l'objet d'un enregistrement sur le système de caisse.**

Plan d'action : Simplification de la fonction showPrintAction qui gère les impressions de ticket, reçus, bons de commande, notes de frais, qui désormais ne filtre plus suivant le type de justificatif imprimé, mais appelle systématiquement la fonction factureEnvoyee

```
function showPrintAction($idcommande=0) {  
    return  
    showPrintActionInt($idcommande)."factureEnvoyee('idcommande=".$idcommande.'");"  
};
```

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Exigence 7 :La documentation relative à l'exigence 7 sur les données cumulatives et récapitulatives ne référence pas les modalités de calcul de ces données par le système de caisse lors de chaque clôture.**

Plan d'action : Ajout d'informations dans dossier d'architecture technique ; 3.4.2.2

Modalités de calcul

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Exigence 9: La notion de ticket sans prix ne fait pas l'objet d'une documentation explicite.**

Plan d'action : Ajout d'informations dans dossier d'architecture technique ; 3.3.

Sécurisation des justificatifs (ajout lien vers documentation et quelques informations directement dans le paragraphe)

<https://kash.click/logiciel-de-caisse-enregistreuse/justificatifsPaiement/>

<https://kash.click/logiciel-de-caisse-enregistreuse/justificatifTkSansPrix/>

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. utilisateur

**- Exigence 17: les archives fiscales sont conservées en base de données au sein de la table RapportFichiers. L'éditeur fait le choix de ne conserver les fichiers dans cette table uniquement durant une période pouvant aller de 10 secondes à 7 ans car le système peut être régénérer à la volée. La documentation ne précise pas les durées de conservation des archives et la justification relative à l'absence de conservation des archives.**

Plan d'action : Ajout d'informations dans dossier d'architecture technique ; 5.1 Données utilisateur

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Exigence 18: La durée maximale pour utiliser le système en mode hors ligne est de 24h, passé ce délai le mode hors ligne n'est plus utilisable, et l'utilisateur sera invité à se connecter à internet pour enregistrer ses données préparées. La durée maximale est acceptable. En revanche, la documentation n'intègre pas de justification explicite pour expliquer ce choix technique, comme le requiert l'exigence 18.**

Plan d'action : Ajout d'informations dans dossier d'architecture technique ; 6. Mode hors ligne

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire

**- Exigence 12 : le mécanisme d'obfuscation des clés est trivial, et ne permet pas de démontrer une obfuscation sécurisé de la clé (la clé est dans une variable, avec un retrait de la dernière clé).**

Plan d'action :

Un mécanisme d'obfuscation de la clé ont été mis en place :

- obfuscation des variables lors de la compilation du code

## Dossier de Maintenance

- utilisation d'un algorithme cryptographique codé sans librairies (cryptographie sur une base XOR avec clé de vérification sha256) afin de crypter la clé privée dans le code source lui-même (code non distribué, seul l'exécutable est distribué).

Ce changement n'implique pas de mise à jour majeure.

### Extrait de code :

```
$encryptedPK = "AFF9LFdDS1BXZxkL*****NgXWcnBhkXaAlnMAJj5A";
function makeLength ($txt,$len)
{
    while(strlen($txt)<$len) {
        $tmp = (string)$txt;
        $txt .= $tmp;
    }
    return substr($txt,0,$len);
}
function myxor($a,$b)
{
    return $a^makeLength($b,strlen($a));
}
function base64_url_decode($input) {
    return base64_decode(strtr($input, '-_', '+/='));
}
function quizzdecode ($txt)
{
    $int = myxor( base64_url_decode(($txt)) , "94OHcur*****OL5Zb6P");

    $tmp1 = substr($int,0,8);
    $tmp2 = myxor( substr($int,8),$tmp1);
    $pos = strpos($tmp2,"_");
    if ($pos>0)
    {
        $length = substr($tmp2,0,$pos+1);
        $str = substr($tmp2,$pos+1,$length);
        $sha = $tmp1;//substr($mix,$pos+1+$length);
        if (substr(sha1($str),0,8)==$sha)
            return $str;
        else
            return "";
    }
    else return "";
}
$encryptedPK = substr($encryptedPK,0,-2);
$encryptedPK .= "PxdONXB1ZgAlKwxETTFaPj0LPWVJSWJ7JmoNMiJeSOVjAD5gPg,, ";

$pk = quizzdecode($encryptedPK);
```

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire  
**- SMC (chapitre 6) : Non-conformité (SMC min) : L'organisation a formalisé un cahier de test mais n'a pas démontré que celui-ci a été déroulé préalablement à l'audit initial.**

Plan d'action : Ajout nouveau chapitre : dossier\_de\_maintenance ; 11. Cahier de tests

Date cible : 30/01/2026 ; Responsable : Simon Cabotse ; Preuve : Doc. réglementaire



## 11. Cahier de tests

Le cahier de test référence la liste des tests effectués avant la publication de versions majeures ainsi que les tests effectués avec une périodicité annuelle.

Ci bas la liste des tests de conformité réalisés :

Test réalisés le 30/01/2026 par Simon Cabotse, sur la version 4.3.151

Création d'un nouveau compte	ok
Enregistrement de ventes	ok
Validation de commandes	ok
Impression de justificatif	ok
Enregistrement de paiements	ok
Répéter	ok x3
Modifier la date système pour la journée suivante	ok
Répéter l'enregistrement de commande d'article et de paiement	ok x 2
En page rapport télécharger un rapport	ok
Télécharger le fichier d'archive fiscale	ok
Vérifier que les signatures cryptographiques sont bien exactes	ok
Altérer le fichier (rapport) par une opération manuelle en modifiant directement les octets du fichier téléchargé	ok

Dossier de Maintenance

Vérifier que la chaîne de signatures cryptographiques n'est plus valide à l'aide de l'outil de vérification des signatures	ok
Modifier directement dans la base de données un champ d'une donnée de vente : ceci pour chaque type de table : Télécharger à nouveau le rapport et constater que les signatures sont bien indiquées comme incorrectes	ok
articles	ok
commande	ok
paiement	ok
fermetureAuto	ok
traçabilité	ok
Repositionner le champ à sa valeur d'origine systématiquement	ok
Indiquer OK si tout les points ont été validés, KO si un échec a été rencontré	ok

## 11. Conformité au Référentiel de Certification

### **Politique de versionnage (Exigence IV.10 - Exigence 21)**

La gestion des versions majeures et mineures est rigoureuse et traçable via SVN.

### **Gestion des mises à jour (Exigence III.6)**

Chaque modification du code est vérifiée pour garantir la conformité continue avant déploiement.

### **Traçabilité des évolutions (Exigence III.13)**

Chaque évolution est documentée, archivée et accessible pour consultation.

### **Surveillance des vulnérabilités (Exigence III.4)**

Un processus de veille de sécurité garantissant la correction rapide des failles détectées.