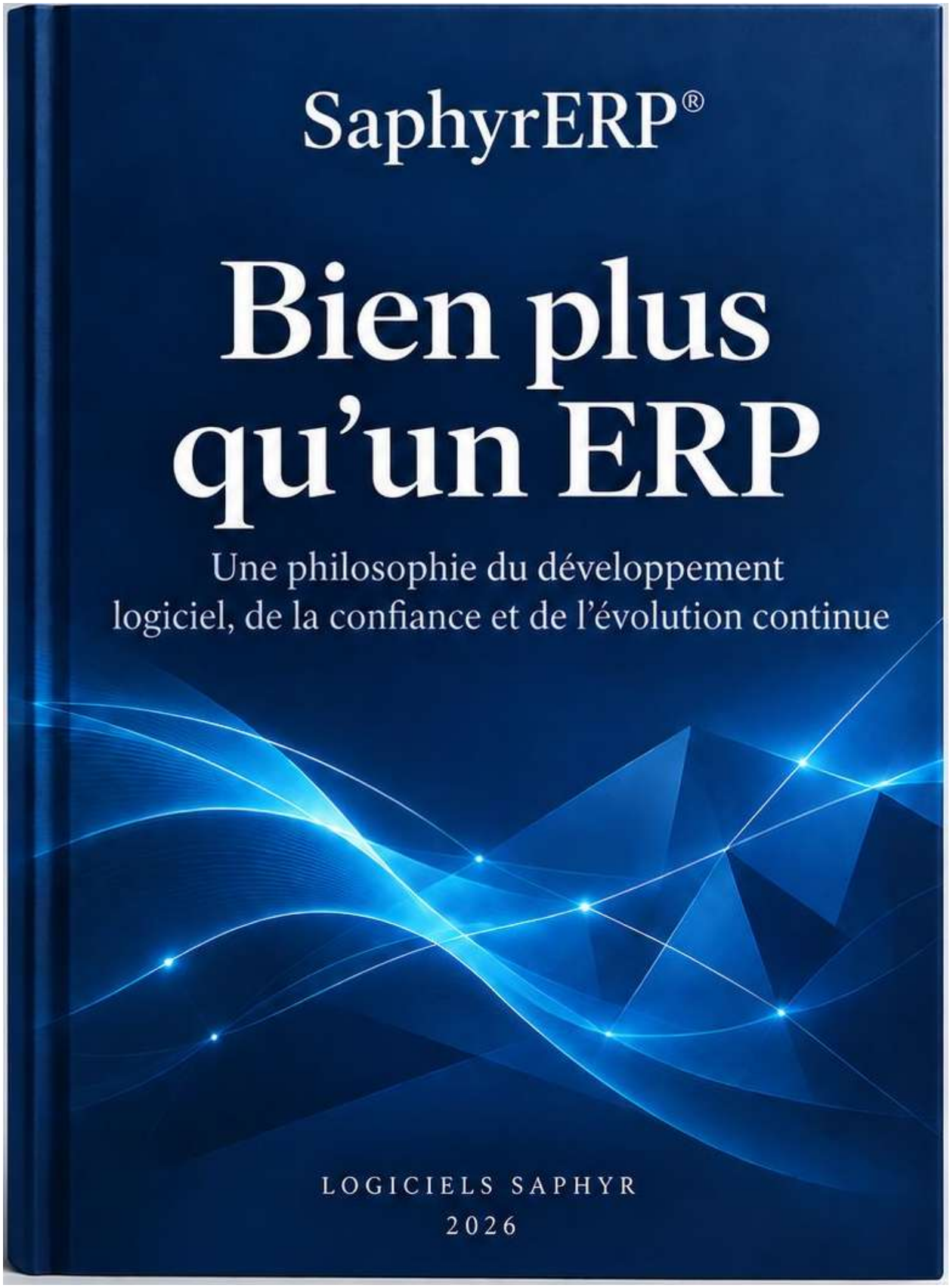
SaphyrERP®

Bien plus qu'un ERP

Une philosophie du développement logiciel, de la confiance et de l'évolution continue



LOGICIELS SAPHYR

2026

SaphyrERP®

Bien plus qu’un ERP

Une philosophie du développement logiciel,
de la confiance et de l’évolution continue

Édition 1.0 · 2026

Dédié à toutes celles et tous ceux qui pensent que les logiciels doivent simplifier le travail, et non le compliquer.

Lettre au lecteur

Cher lecteur,

Si vous avez ouvert ce livre, vous vous attendez probablement à découvrir l'histoire d'un ERP. Vous pourriez être surpris. Le livre que vous tenez entre les mains n'est ni un manuel d'utilisation ni une liste de fonctionnalités. C'est un livre sur les personnes, les entreprises, la confiance et la patience nécessaire pour construire quelque chose qui reste utile pendant de nombreuses années.

Dans le monde du logiciel, on parle beaucoup de technologies, de langages, de frameworks, de cloud, d'automatisation et d'intelligence artificielle. Tous ces sujets sont importants. Mais une question plus simple et plus importante les précède : comment aider les gens à mieux travailler ?

C'est de cette question qu'est né SaphyrERP. Il n'a pas été conçu pour démontrer la sophistication de la technologie, mais pour devenir un outil de travail fiable. Un outil capable de s'adapter à l'entreprise et d'évoluer avec elle.

Dans les pages qui suivent, nous parlons de notre manière de prendre des décisions, de ce que nos clients nous ont appris, de stabilité et de changement, de personnalisation sans perte de cohérence et de la responsabilité qui naît lorsque l'activité d'une entreprise repose, jour après jour, sur le logiciel que l'on construit.

Si, à la fin de ce livre, vous portez un regard un peu différent sur les logiciels, il aura atteint son objectif.

Bienvenue dans l'univers SaphyrERP.

L'équipe Logiciels Saphyr / Safir Informatica

Comment lire ce livre

Ce livre est organisé en trois parties. La première raconte d'où nous venons et quelles valeurs nous ont façonnés. La deuxième explique notre conception du produit et de sa relation avec les utilisateurs. La troisième se tourne vers l'avenir : l'intégration, l'intelligence artificielle, la continuité et la responsabilité de transmettre un produit mature.

Chaque chapitre commence par un principe et se termine par quelques enseignements. Entre les parties, des interludes proposent des idées brèves, pour laisser place à la réflexion. Les détails d'utilisation, les écrans et les procédures techniques relèvent des manuels ; nous conservons ici ce qui doit rester valable même lorsque la technologie change.

Table des matières

Lettre au lecteur 4
Comment lire ce livre 5
Table des matières 6
PARTIE I 7
1. Un début qui ne ressemblait pas à un début 7
2. D'un problème réel à un produit 8
3. Les premiers clients et la première leçon de confiance 9
4. Des applications distinctes à l'idée d'un ERP 10
5. SaphyrERP en-ligne : la même philosophie, une autre façon de travailler 11
PARTIE II 12
6. Comprendre avant de développer 12
7. Les personnes avant le logiciel 13
8. La simplicité est un travail d'ingénierie 14
9. Personnaliser sans dénaturer le produit 15
10. Le logiciel n'est qu'une partie du produit 16
11. L'amélioration continue 17
12. Stabilité, compatibilité et respect de l'investissement du client 18
13. Les décisions qui ont construit SaphyrERP 19
14. Les données, la sécurité et la responsabilité 20
15. L'assistance, là où le produit apprend 21
PARTIE III 22
16. L'intégration, un ERP n'a pas à tout faire 22
17. L'Europe, la facturation électronique et le changement continu 23
18. L'intelligence artificielle comme outil, et non comme finalité 24
19. Le plaisir de construire 25
20. L'avenir ne commence pas par la technologie 26
21. Bien plus qu'un ERP 27
Lettre à l'avenir 28
Chronologie de SaphyrERP 29
Les principes de SaphyrERP 30
Note de l'édition 1.0 31

PARTIE I

D'où nous venons

1. Un début qui ne ressemblait pas à un début

> *« Parfois, un parcours professionnel commence précisément au moment où l'on ose quitter la voie que l'on connaît. »*

L'histoire de SaphyrERP ne commence pas par l'informatique. Elle commence par un changement de direction.

En 1998, après des études à la faculté de médecine de Bucarest, poursuivre dans le domaine médical aurait été le choix naturel. Mais avoir consacré des années à une voie ne signifie pas toujours qu'il faut y rester. Une période a suivi où l'avenir était incertain, mais où sont apparus deux centres d'intérêt qui allaient se rejoindre plus tard dans un même produit : la comptabilité et l'informatique.

La comptabilité a permis de comprendre la logique d'une entreprise : les documents, les flux, les responsabilités, les règles et leurs conséquences. L'informatique a offert la possibilité de transformer cette logique en outils. Ont suivi des cours de HTML, JavaScript, CSS, C++, PHP, Word et Excel, à une époque où l'usage des ordinateurs personnels commençait à se développer rapidement en Roumanie.

Le français, appris par plaisir dès l'enfance, allait devenir un autre atout d'une importance inattendue. En 2001, chez QSP Systems, une entreprise française disposant d'une filiale à Bucarest, les connaissances en comptabilité et en français se sont révélées utiles dans un poste de test logiciel. Le travail au siège de l'entreprise et chez des clients en France a permis un contact direct avec les développeurs, les applications de gestion et la manière dont les logiciels sont utilisés dans la vie réelle.

C'est là que Gupta et SQL sont entrés dans l'histoire et que les bases de données Oracle ont intégré le quotidien. Mais l'enseignement le plus important ne concernait pas une technologie. Il tenait à cette idée : pour comprendre un bon logiciel, il faut adopter le point de vue de celui qui l'utilise.

En 2004, un nouveau changement de direction s'est produit : le départ de l'entreprise et la décision de créer un logiciel de comptabilité plus proche des besoins observés sur le terrain. Une licence Gupta a été achetée avec quelques anciens collègues, et les premières lignes de code ont vu le jour.

Avec le recul, les débuts de SaphyrERP ne ressemblent pas à un plan parfaitement exécuté. Ils apparaissent comme une succession d'expériences différentes qui, à un moment donné, se sont rejointes. La médecine a laissé la rigueur de la recherche des causes, la comptabilité a apporté le langage de l'entreprise, les tests ont développé l'attention aux erreurs et la programmation a fourni l'outil pour construire.

Ce que nous avons appris

- Des expériences apparemment sans lien peuvent devenir un atout pour un produit.
- Comprendre le métier est aussi important que maîtriser la technologie.
- Changer de direction peut être un commencement, et non un abandon.

2. D'un problème réel à un produit

> *« Tout produit durable commence par un problème que quelqu'un a choisi de comprendre en profondeur. »*

En 2004, l'objectif n'était pas de construire un « écosystème ERP ». Il était bien plus concret : créer un logiciel de comptabilité répondant mieux aux besoins rencontrés sur le terrain.

Ce détail compte. Les grands produits n'ont pas besoin de naître avec l'ambition de devenir grands. Il est parfois plus judicieux de commencer par un problème bien défini et la volonté de le résoudre correctement.

À mesure que le logiciel entrait dans les entreprises, les problèmes se diversifiaient. La comptabilité était liée à la facturation. La facturation était liée aux stocks. Les stocks étaient liés aux achats et aux ventes. La paie, les immobilisations, la trésorerie, le CRM, la production et l'analyse financière n'étaient pas des îlots. Ils faisaient partie d'une même réalité.

C'est ainsi qu'est apparue progressivement l'idée qui allait définir SaphyrERP : chaque service ne devrait pas avoir à ressaisir les mêmes informations, et le logiciel devrait suivre les flux réels de l'entreprise.

Au lieu de traiter chaque module comme un logiciel distinct, nous avons commencé à considérer l'entreprise comme un système. Une facture n'est pas seulement un document commercial ; elle peut avoir des effets sur les stocks, la comptabilité, la trésorerie, la relation client et le reporting. Un ERP prend toute sa valeur lorsque ces liens deviennent naturels.

Ce que nous avons appris

- Partir du problème, et non de la liste des fonctionnalités.
- L'intégration des données réduit le travail répétitif et les erreurs.
- Un ERP doit refléter les relations entre les processus, et non l'organigramme du logiciel.

3. Les premiers clients et la première leçon de confiance

« Un client qui reste pendant des années en dit plus sur un produit qu'une démonstration réussie. »

Les premiers clients de Safir sont arrivés en 2005. Trois d'entre eux utilisent encore le produit aujourd'hui.

Cette continuité a pour nous une signification particulière. En deux décennies, les systèmes d'exploitation, les ordinateurs, la législation, les procédures fiscales, les technologies et les attentes ont changé. Les entreprises elles-mêmes ont évolué. Si la relation s'est poursuivie, c'est que le produit n'a pas simplement été acheté : il a été intégré à leur activité.

La confiance ne se construit pas sur une promesse. Elle se construit dans de petits moments : lorsqu'une mise à jour réglementaire arrive à temps, qu'un rapport reste exact, qu'un problème est compris, que les données sont préservées ou qu'un utilisateur peut continuer à travailler.

Dans les logiciels de gestion, la stabilité possède une valeur qui n'est pas toujours visible. Lorsque tout fonctionne, l'utilisateur le remarque à peine. C'est précisément là que réside la réussite.

Nos clients de longue date nous ont aussi appris autre chose : la relation avec un ERP n'est pas une transaction ponctuelle. C'est une collaboration à long terme. De là découle la responsabilité de préserver la compatibilité, d'expliquer les changements et de ne pas transformer chaque nouveauté technologique en rupture pour l'utilisateur.

Ce que nous avons appris

- La confiance se construit par la constance.
- La stabilité est une fonctionnalité, même si elle ne figure pas dans un menu.
- Une relation à long terme change la responsabilité du fournisseur.

4. Des applications distinctes à l'idée d'un ERP

« Un produit devient une plateforme lorsqu'il commence à réunir des problèmes jusque-là résolus séparément. »

Au cours des années suivantes, des logiciels et des modules ont vu le jour dans différents domaines. Chaque projet répondait à un besoin concret. Vers 2018, il est toutefois devenu évident qu'ils pouvaient apporter davantage de valeur en étant réunis dans un système modulaire.

C'est ainsi que SaphyrERP s'est affirmé comme un ERP commercialisé par abonnement mensuel : le client utilise et paie les modules dont il a besoin, pour le nombre d'utilisateurs nécessaire.

Le modèle modulaire a une conséquence importante sur le développement. Le produit doit être suffisamment riche pour des entreprises différentes, sans obliger chaque client à acheter ou à gérer des fonctions qu'il n'utilise pas.

L'abonnement mensuel a été choisi comme modèle commercial, mais aussi comme expression d'une relation continue. La maintenance et les mises à jour standard sont incluses pendant la durée de l'abonnement. Le produit n'est pas abandonné après l'installation ; il continue d'évoluer.

La licence par utilisateur permet d'adapter facilement le modèle lorsque l'équipe du client s'agrandit. De nouveaux modules peuvent être ajoutés sans reconstruire toute l'installation. L'idée centrale reste la même : le produit doit s'adapter à l'entreprise.

Ce que nous avons appris

- La modularité donne de la liberté au client.
- Le modèle commercial doit soutenir le modèle de développement.
- L'évolution du produit fait partie de la valeur de l'abonnement.

5. SaphyrERP en-ligne : la même philosophie, une autre façon de travailler

> *« Parfois, un client ne demande pas une nouvelle fonction. Il demande une autre façon de travailler. »*

En 2013, un projet pour ROMFOUR TUR, entreprise de transport de colis et de transport international de voyageurs, a fait émerger un besoin différent : un accès en ligne, depuis plusieurs lieux et sur des appareils mobiles.

La réponse a été de commencer à réécrire certains modules en PHP, avec le framework Yii. C'est ainsi qu'est né SaphyrERP en-ligne.

La version en ligne n'a pas été conçue comme une simple copie de l'application de bureau dans un navigateur. Avec le temps, elle est devenue un produit à part entière, adapté aux entreprises ayant besoin d'un accès depuis plusieurs sites. Certains modules de la version de bureau — comme la comptabilité, la paie et l'analyse financière — n'ont pas été intégralement réécrits pour la version en ligne, et les deux produits sont restés complémentaires dans certains cas d'usage.

Les bases de données ont des structures proches et peuvent être synchronisées périodiquement. Ce choix préserve un principe important : nous n'imposons pas à toutes les entreprises de travailler de la même manière.

Pour certains clients, une base locale et une application de bureau sont la solution la plus adaptée. Pour d'autres, l'accès web est essentiel. Parfois, la bonne solution combine les deux.

Ce que nous avons appris

- Il n'existe pas d'architecture unique adaptée à tous.
- Passer en ligne ne doit pas signifier copier mécaniquement l'application de bureau.
- Des produits complémentaires peuvent préserver la liberté de choix du client.

« Nous ne construisons pas pour la technologie. Nous construisons pour le travail. »

PARTIE II

Notre conception du produit

6. Comprendre avant de développer

> *« Une demande est le début d'une conversation, et non le cahier des charges définitif. »*

Lorsqu'un client demande une fonctionnalité, notre premier réflexe n'est pas d'ouvrir l'éditeur de code. Nous essayons d'abord de comprendre pourquoi il en a besoin.

Parfois, la demande décrit la solution imaginée par l'utilisateur, plutôt que le problème. Si nous en comprenons la cause, nous pouvons trouver une solution plus simple, plus sûre ou plus générale. Et si cette solution peut aussi être utile à d'autres clients, elle peut intégrer le produit standard.

Cette manière de travailler demande de la patience. Il est plus rapide de réaliser exactement ce qui est demandé. Mais cette rapidité initiale peut créer de la complexité à long terme : des options presque identiques, des exceptions difficiles à maintenir et des flux que plus personne ne comprend quelques années plus tard.

C'est pourquoi nous essayons de poser des questions : quel résultat recherchez-vous ? Qui utilise l'information ? Que se passe-t-il avant et après ? Existe-t-il déjà une fonction qui résout une partie du problème ? La solution peut-elle être généralisée sans devenir lourde ?

Cette discipline est l'une des formes les plus importantes de respect du produit.

Ce que nous avons appris

- Le besoin exprimé ne correspond pas toujours au problème réel.
- Une généralisation bien pensée réduit la complexité future.
- Comprendre avant de programmer fait gagner du temps à long terme.

7. Les personnes avant le logiciel

« Avant de concevoir un écran, il faut comprendre la personne qui va l'utiliser. »

Un même ERP est utilisé par des personnes dont les rôles, les rythmes et les objectifs diffèrent. Un opérateur veut saisir rapidement les données. Un comptable recherche l'exactitude et la traçabilité. Un dirigeant souhaite une vue synthétique. Un administrateur système pense aux droits d'accès et à la sécurité.

Un bon produit n'oblige pas tout le monde à penser comme le développeur. Il traduit la complexité interne en un processus de travail compréhensible.

C'est pourquoi l'expérience utilisateur ne se résume pas à des couleurs et à des boutons modernes. Dans un ERP, une bonne expérience repose sur un enchaînement logique des étapes, des saisies automatiques correctes, des contrôles au bon moment, des messages clairs et la possibilité de trouver rapidement l'information.

Elle implique aussi de respecter les habitudes acquises. Un utilisateur qui travaille depuis des années avec une application y a investi du temps. Une modernisation qui ignore cet investissement peut créer plus de frustration que de valeur.

Nous modernisons lorsqu'il existe un bénéfice réel, et non simplement parce qu'une nouvelle interface est plus belle sur une capture d'écran.

Ce que nous avons appris

- Dans un ERP, l'UX concerne avant tout le déroulement du travail.
- Des rôles différents demandent des points de vue différents sur les mêmes données.
- La modernisation doit respecter le temps investi par les utilisateurs.

8. La simplicité est un travail d'ingénierie

> *« Il est facile d'ajouter. Il est bien plus difficile de garder les choses simples. »*

Un produit mature accumule inévitablement des fonctions. Chaque client peut apporter une bonne idée, chaque évolution réglementaire peut imposer un nouveau champ, chaque intégration peut ajouter des paramètres.

Le danger est que la maturité se transforme en encombrement.

La simplicité ne signifie pas avoir peu de fonctions. Elle signifie que l'utilisateur voit, au bon moment, exactement ce dont il a besoin. Elle repose sur des valeurs par défaut intelligentes, la réutilisation des informations existantes et la suppression des étapes qui n'apportent aucune valeur.

Parfois, une amélioration importante est un champ rempli automatiquement. D'autres fois, c'est un filtre mémorisé, un contrôle ou un bouton déplacé à un endroit plus naturel. Quelques secondes gagnées sur une opération répétée des centaines de fois peuvent valoir davantage qu'une fonction spectaculaire utilisée une fois par mois.

C'est pourquoi, dans SaphyrERP, l'amélioration consiste souvent à réduire : moins de clics, moins de doubles saisies, moins de situations où l'utilisateur doit mémoriser ce que le logiciel peut déjà savoir.

Ce que nous avons appris

- La simplicité n'est pas l'absence de fonctionnalités.
- Automatiser de petites tâches répétitives a un impact considérable.
- Chaque nouvelle option doit justifier la complexité qu'elle ajoute.

9. Personnaliser sans dénaturer le produit

« Un ERP doit s'adapter au client sans devenir un produit entièrement différent pour chacun. »

La personnalisation a toujours fait partie de SaphyrERP. Les entreprises ont des processus différents, et un ERP rigide finit soit par être contourné, soit par obliger l'organisation à travailler de manière artificielle.

Mais la personnalisation comporte aussi un risque : si chaque demande devient une branche distincte, le produit se fragmente. Les mises à jour deviennent difficiles, les tests se multiplient et l'expérience acquise ne peut plus être transmise d'un client à l'autre.

C'est pourquoi nous essayons de distinguer trois situations. Premièrement, le problème peut être résolu par le paramétrage. Deuxièmement, la solution est suffisamment générale pour intégrer le produit standard. Troisièmement, le besoin est réellement spécifique et justifie un développement sur mesure.

Même dans ce dernier cas, nous recherchons des composants réutilisables et évitons, autant que possible, les modifications qui rompent la compatibilité avec le produit standard.

Une bonne personnalisation n'est pas du code ajouté au hasard. C'est une extension maîtrisée.

Ce que nous avons appris

- Le paramétrage doit être préféré au code lorsqu'il suffit.
- Une personnalisation doit aussi être évaluée au regard de son coût de maintenance futur.
- Le produit standard reste au centre de l'écosystème.

10. Le logiciel n'est qu'une partie du produit

« Un ERP ne s'arrête pas à l'installation. C'est alors que commence la relation avec le client. »

Lorsqu'on découvre un ERP, on voit des écrans, des menus et des rapports. Mais ils n'en représentent que la partie visible.

Autour de SaphyrERP existe un écosystème : le site commercial, le Portail Client, la gestion des licences en ligne, les mises à jour, l'assistance par e-mail et à distance, les tickets, la synchronisation, la facturation des abonnements et les intégrations avec des services externes.

Les modules se commandent en ligne. Après paiement, le programme d'installation peut être téléchargé et l'activation de la licence s'effectue en ligne. Au démarrage, l'application vérifie la validité de la licence et l'existence d'une nouvelle version. En l'absence de connexion Internet, elle peut continuer à fonctionner ; la vérification reprend à la prochaine ouverture avec connexion.

L'ajout d'un module ou d'un utilisateur ne devrait pas imposer de réinstaller tout le système. Après synchronisation des droits et réouverture du logiciel, la nouvelle configuration peut devenir disponible.

Le Portail Client donne accès aux liens d'installation et de mise à jour, aux factures et à l'ouverture de tickets. Les abonnements sont facturés chaque mois dans SaphyrERP et synchronisés périodiquement avec le portail, où ils peuvent être réglés par carte bancaire.

Tout cela ne constitue pas des services « autour du produit ». C'est le produit.

Ce que nous avons appris

- L'expérience client couvre tout le parcours, et pas seulement l'application.
- La gestion des licences et les mises à jour doivent être aussi peu intrusives que possible.
- Le portail, l'assistance et la facturation font partie de l'écosystème.

11. L'amélioration continue

> *« Un produit ne devient pas mature en un jour. Il le devient en ne cessant jamais d'évoluer. »*

SaphyrERP n'est jamais « terminé ». Non parce qu'il serait incomplet, mais parce que le monde dans lequel il est utilisé ne s'arrête pas.

La législation change. Les clients grandissent. De nouveaux services apparaissent. Des processus manuels hier peuvent être automatisés demain. Un produit qui refuse d'évoluer devient progressivement une contrainte.

L'amélioration continue ne signifie toutefois pas modifier constamment ce qui est familier. Parfois, la meilleure évolution est interne : un algorithme plus sûr, un import plus robuste, un meilleur contrôle, une requête plus rapide.

D'autres fois, il faut reconnaître qu'une technologie a atteint ses limites et préparer une transition. Évoluer de manière responsable, c'est construire sur le passé plutôt que le nier.

Nous ne recherchons pas la perfection. Nous recherchons un progrès constant.

Ce que nous avons appris

- Un ERP mature continue d'apprendre.
- Le changement visible n'est pas la seule forme de progrès.
- Une évolution responsable préserve ce qui fonctionne et modifie ce qui doit l'être.

12. Stabilité, compatibilité et respect de l'investissement du client

« La stabilité n'est pas l'absence de changement. C'est un changement mené avec soin. »

Dans un produit utilisé quotidiennement, une modification apparemment mineure peut avoir des conséquences importantes. Un champ renommé peut perturber une habitude. Une nouvelle règle peut modifier un rapport. Une mise à jour technique peut interagir avec l'infrastructure du client.

La compatibilité est donc une préoccupation constante. Nous essayons d'éviter les ruptures inutiles et de préserver la continuité des données et des processus de travail.

SaphyrERP Desktop utilise des bases de données Firebird. La base peut être installée sur l'ordinateur du client, sur un serveur du réseau local ou sur un VPS Windows, y compris pour un accès via VPN. Cette souplesse permet aux entreprises de choisir l'infrastructure qui leur convient.

Mais la souplesse s'accompagne de responsabilités. Une application de gestion vit dans un écosystème de systèmes d'exploitation, de pilotes, de réseaux, de droits d'accès, de sauvegardes et de services externes. Tout changement doit être évalué dans ce contexte, et non isolément.

Respecter l'investissement du client, c'est éviter de lui demander de reconstruire inutilement ce qui fonctionne.

Ce que nous avons appris

- La compatibilité est une forme de respect.
- L'infrastructure réelle du client compte dans la conception du produit.
- Une migration doit être justifiée par la valeur qu'elle apporte, et non par la mode.

13. Les décisions qui ont construit SaphyrERP

« L'architecture d'un produit raconte l'histoire de ses décisions. »

Application de bureau et Firebird

L'application de bureau répondait à une réalité où la rapidité en local, la maîtrise des données et le fonctionnement en réseau étaient essentiels. Firebird a fourni une base de données relationnelle adaptée à ce modèle, avec des installations locales, des serveurs LAN et des VPS Windows.

Un abonnement plutôt qu'une licence perpétuelle

L'abonnement mensuel associe le droit d'utilisation à la maintenance et aux mises à jour. Le client paie tant qu'il utilise le produit, pour les modules et les utilisateurs dont il a besoin. De son côté, le fournisseur s'engage à assurer la continuité du produit.

Application de bureau et version en ligne, sans migration forcée

L'arrivée de SaphyrERP en-ligne n'a pas entraîné l'abandon de l'application de bureau. Les deux modèles répondent à des besoins différents et peuvent coexister. Ce choix évite de transformer une tendance technologique en obligation pour le client.

Intégrer plutôt que réinventer

Pour les paiements, l'Open Banking ou la facturation électronique, nous choisissons l'intégration avec des prestataires spécialisés lorsque c'est la meilleure solution. Un ERP n'a pas besoin de devenir une banque, un prestataire de paiement par carte ou une plateforme de facturation électronique pour offrir à l'utilisateur un parcours cohérent.

Une gestion des licences en ligne tolérant les coupures temporaires d'Internet

La licence est vérifiée en ligne, mais une interruption temporaire de la connexion ne doit pas bloquer inutilement l'activité. La vérification peut reprendre au prochain démarrage avec connexion.

Ce que nous avons appris

- Chaque choix technique doit répondre à un besoin réel.
- Une bonne architecture laisse place à plusieurs façons de travailler.
- L'intégration est souvent plus judicieuse que la reconstruction d'un service spécialisé.

14. Les données, la sécurité et la responsabilité

« Les données d'une entreprise ne sont pas un accessoire de l'application. Elles font partie de sa mémoire. »

Un ERP collecte et traite des informations importantes : des données sur les clients, les fournisseurs, les salariés, les transactions, les documents et les processus internes. Notre manière de traiter les données est donc indissociable de notre manière de construire le produit.

Dans les installations de bureau, la base Firebird se trouve dans l'infrastructure choisie par le client. Celui-ci en garde ainsi la maîtrise, mais assume aussi la responsabilité des accès, des sauvegardes, du système d'exploitation, du réseau et de la protection du serveur.

De notre côté, les accès nécessaires à l'assistance doivent être limités au strict nécessaire et effectués avec l'accord du client. Les données ne doivent pas devenir une ressource commerciale secondaire.

La sécurité n'est pas un bouton que l'on active à la fin. C'est une succession de choix : droits par utilisateur, mots de passe, protection des connexions, mises à jour, sauvegardes, journalisation et procédures de gestion des incidents.

La confiance dans un ERP repose aussi sur la certitude que les données restent disponibles, compréhensibles et sous le contrôle légitime de l'entreprise.

Ce que nous avons appris

- Les données appartiennent au client.
- La sécurité est une responsabilité partagée.
- La sauvegarde et le contrôle des accès sont des processus, et non de simples paramètres.

15. L'assistance, là où le produit apprend

« Un ticket d'assistance peut être le point de départ de la prochaine amélioration. »

L'assistance est souvent considérée comme une activité distincte du développement. Pour nous, elle constitue l'une des principales sources d'information sur le produit.

Un message d'erreur récurrent révèle les endroits où l'interface n'explique pas suffisamment les choses. Une question fréquente peut signaler un processus trop complexe. Une intervention à distance peut mettre en évidence l'écart entre le scénario imaginé pendant le développement et la réalité de l'entreprise.

C'est pourquoi l'assistance par e-mail, par tickets et à distance ne se limite pas à la résolution d'incidents. Elle constitue un retour d'expérience structuré.

Par ailleurs, une bonne assistance doit éviter de créer une dépendance. Si la même question revient souvent, la solution n'est pas de répondre plus vite à chaque fois, mais d'améliorer le produit, le message ou la documentation.

Le meilleur ticket est parfois celui qui n'aura plus besoin d'être ouvert après la prochaine version.

Ce que nous avons appris

- L'assistance nourrit la conception du produit.
- Les problèmes récurrents doivent être résolus à la racine.
- Une documentation et des messages clairs réduisent les besoins d'intervention.

« La stabilité est ce que l'utilisateur ne remarque que lorsqu'elle fait défaut. »

PARTIE III

Où nous allons

16. L'intégration, un ERP n'a pas à tout faire

« L'intégration est une forme de respect des compétences des autres. »

Dans l'écosystème numérique moderne, aucune application sérieuse ne fonctionne seule. Les paiements sont traités par des prestataires spécialisés. Les banques donnent accès aux données par des services réglementés. La facturation électronique passe par des plateformes et des normes. Les transporteurs, les boutiques en ligne et les administrations ont leurs propres systèmes.

Il serait possible de tenter de reconstruire une partie de ces services. Mais la bonne question est de savoir si cela apporte de la valeur au client.

Nous préférons investir dans l'intégration : faire circuler correctement l'information, éviter les ressaisies manuelles et conserver SaphyrERP comme un point de travail cohérent.

Les paiements en ligne, l'import des relevés bancaires, les services de facturation électronique et les API illustrent la même philosophie. Nous ne voulons pas remplacer les prestataires spécialisés. Nous voulons éviter que l'utilisateur soit obligé de recréer manuellement le lien entre eux.

Un ERP moderne est, de plus en plus, un orchestrateur de l'information.

Ce que nous avons appris

- Il n'est pas nécessaire de développer chaque service en interne.
- De bonnes API réduisent le travail manuel.
- La valeur de l'ERP augmente lorsqu'il relie de manière cohérente des services spécialisés.

17. L'Europe, la facturation électronique et le changement continu

« La législation change. Le rôle du produit est de transformer ce changement en un processus de travail maîtrisable. »

Le développement de SaphyrERP en France a apporté un nouveau regard sur l'adaptation. Traduire l'interface ne suffit pas. Il faut comprendre les règles, les documents, les pratiques et l'infrastructure numérique d'un autre marché.

En France, SaphyrERP est commercialisé par LOGICIELS SAPHYR. Pour la facturation électronique, l'orientation retenue est celle d'une solution compatible, intégrée par API à une plateforme spécialisée, plutôt que de transformer SaphyrERP lui-même en plateforme de distribution.

L'intégration avec SuperPDP illustre cette approche : SaphyrERP reste l'environnement de travail de l'utilisateur, tandis que le service spécialisé assure les fonctions de la plateforme externe. Pour le client, la complexité technique doit rester autant que possible en arrière-plan du processus de travail.

La même idée s'applique en Roumanie ou dans d'autres pays de l'Union européenne. Les exigences fiscales peuvent différer, mais le principe reste le même : le produit doit s'adapter sans perdre sa cohérence.

L'internationalisation ne se résume pas à la traduction. C'est une localisation fonctionnelle.

Ce que nous avons appris

- Un nouveau marché exige une adaptation, et pas seulement une traduction.
- Les exigences réglementaires doivent s'intégrer dans des processus aussi simples que possible pour l'utilisateur.
- L'intégration avec des plateformes spécialisées permet au produit de rester concentré sur sa mission.

18. L'intelligence artificielle comme outil, et non comme finalité

« Une nouvelle technologie mérite d'être adoptée lorsqu'elle résout mieux un problème, et non simplement parce qu'elle est nouvelle. »

L'intelligence artificielle ouvre de réelles possibilités aux logiciels de gestion : recherche dans la documentation, assistance contextuelle, classification, analyse, automatisation de certaines tâches répétitives et soutien à la relation client.

Pour SaphyrERP, la question n'est pas « où pouvons-nous mettre de l'IA ? », mais « où l'IA peut-elle réduire le travail sans réduire la maîtrise ? ».

Un assistant SaphyrERP peut aider l'utilisateur à trouver rapidement une fonction, à comprendre un message ou à parcourir la documentation. Un agent commercial peut aider à organiser les informations de prospection. D'autres outils peuvent proposer, vérifier ou synthétiser.

Mais les décisions comptables, fiscales et opérationnelles importantes doivent rester vérifiables. L'IA doit signaler l'incertitude, permettre le contrôle humain et ne pas transformer une suggestion en vérité simplement parce qu'elle est formulée de manière convaincante.

L'expérience accumulée en deux décennies de processus de gestion constitue précisément un atout à cet égard : elle nous aide à déterminer ce qui mérite d'être automatisé et ce qui doit rester sous contrôle explicite.

Ce que nous avons appris

- L'IA doit résoudre des problèmes, et non décorer le produit.
- La maîtrise et la vérifiabilité sont essentielles dans les logiciels de gestion.
- Avec l'IA, l'expérience métier devient plus précieuse, et non moins précieuse.

19. Le plaisir de construire

« Certains produits sont entretenus. D'autres continuent d'être construits. »

Avec le recul, nous pouvons compter les modules, les versions, les intégrations et les années. Mais ce sont les résultats. Derrière eux, il y a toujours eu la curiosité de comprendre comment mieux faire.

Le logiciel est aussi un savoir-faire artisanal. On revient sur une solution. On l'affine. On découvre un cas que l'on n'avait pas prévu. On simplifie. On optimise. On apprend.

Nous ne construisons pas pour les démonstrations. Une fonction qui impressionne pendant une présentation d'une heure peut valoir moins qu'une autre qui fait gagner quelques minutes chaque jour pendant dix ans.

L'une des plus belles réactions qu'un développeur puisse recevoir n'est pas « c'est spectaculaire », mais « maintenant, c'est plus facile ».

C'est peut-être cette satisfaction qui explique pourquoi, après plus de vingt ans, SaphyrERP continue d'accueillir des idées, des améliorations et de nouveaux projets.

Ce que nous avons appris

- L'utilité au quotidien compte davantage que l'effet produit lors d'une démonstration.
- Un bon logiciel s'affine avec le temps.
- La curiosité est le moteur de l'évolution à long terme.

20. L'avenir ne commence pas par la technologie

« La technologie change les outils. Les valeurs décident de la direction. »

Il est impossible de savoir exactement à quoi ressemblera SaphyrERP dans dix ou vingt ans. Certaines technologies actuelles disparaîtront. D'autres deviendront invisibles, tellement courantes que nous ne les nommerons même plus.

Nous pouvons toutefois dire ce que nous voulons préserver.

Nous voulons que le produit respecte les données du client. Qu'il évolue sans transformer chaque changement en rupture. Qu'il puisse collaborer avec d'autres systèmes. Qu'il garde les personnes au cœur des décisions. Qu'il choisisse la stabilité lorsque la nouveauté n'apporte aucune valeur, et le changement lorsque l'ancienne façon de travailler devient un obstacle.

Nous voulons qu'un client puisse commencer avec ce dont il a besoin, puis ajouter des modules et des utilisateurs. Nous voulons que l'information circule plus facilement entre les applications. Nous voulons que l'automatisation élimine le travail répétitif, et non la responsabilité.

Et nous voulons que l'équipe qui poursuivra le développement du produit comprenne non seulement comment le code est écrit, mais aussi pourquoi certaines décisions ont été prises.

L'avenir n'est pas une version. C'est la capacité de continuer.

Ce que nous avons appris

- Nous ne pouvons pas prédire la technologie, mais nous pouvons préserver les principes.
- L'avenir du produit dépend de sa capacité d'adaptation.
- Les raisons qui ont guidé les décisions doivent être transmises.

21. Bien plus qu'un ERP

> *« Un produit devient plus que la somme de ses fonctions lorsque les personnes peuvent s'appuyer sur lui pour construire en confiance. »*

Le titre de ce livre n'affirme pas que SaphyrERP doit tout faire. Bien au contraire.

« Bien plus qu'un ERP » signifie que le produit ne se réduit pas à des modules, des menus et des rapports. Il incarne une relation à long terme avec les utilisateurs, une manière de prendre des décisions, un écosystème de services et une responsabilité envers les informations que les entreprises confient au système.

C'est poser des questions avant de programmer. Préférer une solution générale lorsqu'elle est plus judicieuse. Personnaliser lorsque la différence compte réellement. Intégrer des services meilleurs que ceux que l'on pourrait construire seul. Préserver la maîtrise du client sur ses données. Ne pas changer uniquement pour paraître moderne.

C'est aussi accepter que le produit ne sera jamais achevé. Tant que les façons de travailler évolueront, que la législation changera et que la technologie créera de nouvelles possibilités, il restera quelque chose à améliorer.

SaphyrERP a commencé en 2004 avec une idée concrète et a grandi à travers des milliers de petites décisions. S'il existe un fil conducteur entre elles, c'est la volonté de rendre le logiciel utile, compréhensible et digne de confiance.

C'est peut-être, au fond, le sens le plus simple du titre : un ERP gère des processus. Un produit auquel on fait confiance devient partie intégrante de la manière dont on construit son entreprise.

Ce que nous avons appris

- Le produit, c'est aussi la relation construite autour de lui.
- La confiance se gagne par des décisions répétées, et non par des slogans.
- Bien plus qu'un ERP signifie davantage de responsabilité, et non davantage de complexité.

« Certains logiciels gèrent des données.
Nous souhaitons construire un logiciel qui mérite la confiance de celles et ceux qui bâtissent leur avenir avec son aide. »

Lettre à l'avenir

« Laisse le produit un peu meilleur que tu ne l'as trouvé. »

Cher collègue,

Si tu lis ces lignes, c'est que tu as rejoint l'équipe qui développe SaphyrERP.

Tu trouveras probablement dans le code des décisions que tu ne comprendras pas immédiatement. C'est normal. Parfois, derrière une ligne de code se cache un problème résolu bien des années auparavant. D'autres fois, tu découvriras que le monde a changé et qu'il est temps de trouver une meilleure solution.

Avant de modifier quelque chose, essaie de comprendre pourquoi cela existe.

Ne conserve pas le passé uniquement parce qu'il est ancien. Mais ne le change pas non plus uniquement parce qu'il est ancien.

Demande-toi quel problème il résout. Examine les données. Écoute l'utilisateur. Cherche la solution simple. Pense aux autres clients et au collègue qui maintiendra le code après toi.

Respecte le travail de ceux qui ont construit avant toi, mais n'aie pas peur de l'améliorer.

Les technologies que nous utilisons aujourd'hui deviendront inévitablement anciennes. Les bons principes durent plus longtemps.

Si tu préserves la curiosité, le respect de l'utilisateur, l'attention portée aux données et la volonté de simplifier, l'esprit de SaphyrERP continuera d'exister, quels que soient le langage, la base de données ou la plateforme.

Laisse le produit un peu meilleur que tu ne l'as trouvé.

— L'équipe SaphyrERP

Chronologie de SaphyrERP

Année	Événement
1998	Fin des études à la faculté de médecine de Bucarest et choix d'une nouvelle voie professionnelle.
2001	Début de l'activité chez QSP Systems : tests logiciels, comptabilité, français, projets en France ; apprentissage de Gupta et de SQL.
2004	Lancement d'une activité indépendante dans le logiciel et début du développement du produit qui deviendra SaphyrERP.
2005	Premiers clients. Trois d'entre eux utilisent encore le produit en 2026.
2013	Début de SaphyrERP en-ligne, en PHP/Yii, pour répondre aux besoins de travail depuis plusieurs sites de ROMFOUR TUR.
2018	Regroupement des applications et des modules dans un ERP modulaire commercialisé par abonnement mensuel.
2026	Développement en France par l'intermédiaire de LOGICIELS SAPHYR ; développement des intégrations de paiement, d'Open Banking et de facturation électronique ; début de l'intégration des agents et assistants IA.

Les principes de SaphyrERP

Comprendre avant de développer. Une demande est le début de l'analyse, et non son aboutissement.

Le logiciel doit s'adapter à l'entreprise. Nous n'obligeons pas les processus réels à se conformer artificiellement à l'application.

La simplicité est un travail d'ingénierie. Nous réduisons les étapes, les doublons et la complexité inutile.

La stabilité est une fonctionnalité. Nous modifions avec soin ce qui soutient l'activité quotidienne.

Les données appartiennent au client. La technologie doit préserver la maîtrise légitime de l'information.

Personnaliser sans fragmenter le produit. Nous recherchons d'abord une solution générale et réutilisable.

Intégrer ce que d'autres font mieux. L'ERP doit orchestrer, et non réinventer chaque service.

L'assistance fait partie du développement. Les questions et les problèmes des utilisateurs nous montrent ce qui doit être amélioré.

Automatiser le travail répétitif, et non la responsabilité. Les personnes doivent pouvoir comprendre et contrôler les décisions importantes.

Évoluer en permanence. Nous ne recherchons pas la perfection, mais un progrès constant.

Note de l'édition 1.0

Cette première édition réunit l'histoire, les principes et les orientations de SaphyrERP après plus de deux décennies de développement. Elle n'est pas une photographie définitive. Comme le produit, ce livre peut évoluer.

C'est pourquoi la dernière page ne porte pas le mot « Fin ».

Version 1.0