

FASB ASU 2025-06 : Guide de capitalisation des logiciels selon l'ASC 350-40

Publié le 4 mai 2026 42 min de lecture



Résumé analytique

La norme FASB ASU 2025-06 introduit des **améliorations ciblées** à l'ASC 350-40 **Logiciels destinés à un usage interne**, modifiant fondamentalement la manière dont les entreprises déterminent le moment où elles doivent comptabiliser les coûts de développement de logiciels en immobilisations. En particulier, l'ASU 2025-06 **supprime le modèle historique fondé sur les étapes de projet** au profit d'un « **seuil de comptabilisation** » unique, basé sur l'approbation de la direction et la probabilité d'achèvement du projet (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Selon les nouvelles directives, les entreprises commencent à comptabiliser les coûts en immobilisations *uniquement lorsque* (1) la direction a autorisé et engagé le financement du projet et (2) il est **probable** que le projet soit achevé et utilisé comme prévu (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Il est crucial de noter que si **une incertitude significative liée au développement** subsiste — telle que des fonctionnalités inédites non résolues ou des exigences changeantes — la comptabilisation doit être différée jusqu'à ce que cette incertitude soit levée (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com). En effet, l'ASU 2025-06 aligne les PCGR (GAAP) sur le développement moderne et itératif (par exemple, la méthode Agile) en supprimant les étapes rigides, tout en conservant les critères fondamentaux de comptabilisation économique.

Pour les **entreprises SaaS**, qui comptabilisent traditionnellement la plupart des coûts de développement de produits selon les **règles relatives aux logiciels destinés à un usage interne**, cette mise à jour peut avoir une incidence significative sur l'information financière. De nombreuses entreprises SaaS ont historiquement comptabilisé d'importantes dépenses de R&D interne en immobilisations, mais devaient naviguer avec précaution dans les règles par étapes de l'ASC 350-40. Selon le nouveau modèle, les entreprises doivent faire preuve de jugement dès le début et documenter le moment où le seuil de « probabilité d'achèvement » est atteint. Si les incertitudes substantielles sont rapidement levées, la comptabilisation peut commencer plus tôt que sous l'ancien modèle ; à l'inverse, les projets ambigus sont entièrement passés en charges jusqu'à ce que les ambiguïtés soient dissipées (Source: www.eidebailly.com) (Source: www.armanino.com). Par exemple, une entreprise **mettant en œuvre un ERP tiers** peut commencer la comptabilisation dès la signature d'un contrat et l'absence de fonctionnalité inédite (Source: www.eidebailly.com), tandis qu'un projet d'application mobile aux exigences indéfinies reste passé en charges jusqu'à ce que la conception soit finalisée (Source: www.eidebailly.com).

Ce rapport fournit une **analyse complète** de l'ASU 2025-06, de son contexte, de sa justification et de ses dispositions détaillées. Nous examinons le **champ d'application et la transition** des nouvelles règles, les comparons aux directives antérieures et aux règles applicables aux logiciels commercialisés en externe, et analysons l'impact sur les états financiers des développeurs SaaS. Le rapport traite également de la manière de mettre en œuvre ces exigences dans la pratique — en se concentrant sur les systèmes comptables tels que **NetSuite** — et présente des études de cas, des enquêtes sectorielles et des commentaires d'experts. Il convient de noter que les principaux cabinets comptables (par exemple, Deloitte, RSM, Baker Tilly, Crowe) soulignent tous que si l'ASU 2025-06 ne modifie pas les coûts qui *sont admissibles* à une comptabilisation en immobilisations, elle supprime les phases séquentielles et repose sur le jugement concernant la viabilité du projet (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Les premières données issues d'une enquête d'Armanino montrent que la plupart des entreprises SaaS comptabilisent déjà les coûts de développement en immobilisations (69 % des meilleures entreprises SaaS l'ont fait en 2024 (Source: www.armanino.com) et que les nouvelles règles codifient simplement une évolution déjà en cours dans la pratique.

Enfin, ce rapport examine les **directives spécifiques à NetSuite** pour les développeurs SaaS. Il décrit comment un fournisseur SaaS peut configurer les fonctionnalités de comptabilité de projet et d'amortissement de NetSuite pour suivre les projets de logiciels internes, allouer les coûts entre immobilisations et charges, et se conformer aux directives mises à jour du FASB. Des exemples pratiques (par exemple, une mise à niveau de plateforme pluriannuelle ou [le développement de fonctionnalités basées sur l'IA](#)) illustrent les jugements requis. Nous examinons également des questions connexes telles que le [traitement fiscal](#) (notamment l'impact de l'IRC §174 sur la comptabilisation de la R&D (Source: www.armanino.com) et les attentes réglementaires. Tout au long du rapport, l'analyse s'appuie sur de nombreuses citations de sources faisant autorité : communiqués du FASB, publications de cabinets comptables, documents et commentaires déposés auprès de la SEC, et enquêtes sectorielles. L'objectif est de doter les dirigeants financiers et technologiques d'une compréhension approfondie de l'ASU 2025-06, des nuances de l'ASC 350-40 et des meilleures pratiques de mise en œuvre dans NetSuite et au-delà.

Introduction et contexte

Le développement de logiciels dans l'entreprise moderne

Dans l'économie actuelle axée sur la technologie, les **logiciels développés en interne** sont devenus un actif essentiel pour les entreprises, en particulier pour les entreprises de **SaaS (logiciel en tant que service)**. Les applications développées en interne — qu'elles servent à gérer les opérations, l'analyse ou les plateformes cloud — peuvent représenter un investissement substantiel, atteignant souvent des millions de dollars. En conséquence, les PCGR américains fournissent des règles sur le moment où il convient de **comptabiliser en immobilisations** ces coûts de développement (les traiter comme des actifs à long terme au bilan) par rapport à celui où il faut les **passer en charges** immédiatement. Historiquement, les PCGR américains établissaient une distinction nette entre les **logiciels destinés à un usage interne** (couverts par l'ASC 350-40) et les **logiciels destinés à être vendus/commercialisés en externe** (ASC 985-20). L'ASC 350-40 a été établie pour la première fois en 2009 (via le FASB ASU 2009-13) pour normaliser le traitement des coûts des logiciels internes. Elle exigeait une comptabilisation en immobilisations pendant la phase de « développement des applications », mais imposait de passer en charges les coûts des phases de planification préliminaire et de post-mise en œuvre (Source: www.cohenco.com). Cette approche par étapes reposait sur un modèle de développement traditionnel en « cascade » : une fois qu'une entité avait terminé la planification et engagé les fonds, les coûts de développement pouvaient être comptabilisés en immobilisations (Source: www.cohenco.com) (Source: www.eidebailly.com).

Au fil du temps, cependant, les pratiques de développement de logiciels ont évolué. De nombreuses entreprises utilisent désormais des **méthodologies itératives ou Agile**, qui ne suivent pas de phases séquentielles distinctes. Comme de nombreux praticiens l'ont noté, sous l'ancienne directive, il pouvait être difficile ou subjectif de déterminer avec précision la transition entre la planification et le développement lorsque le travail est incrémental (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Le FASB lui-même a reconnu ces préoccupations lors de ses recherches sur le terrain : de nombreux préparateurs estimaient que le modèle par étapes de l'ASC 350-40 était *obsolète* compte tenu des méthodes modernes (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Parallèlement, la croissance du [cloud computing](#) et du SaaS a modifié le moment et la manière dont les logiciels sont fournis. Pour les entreprises SaaS, les coûts de développement de produits finissaient souvent par être comptabilisés selon la norme 350-40 (usage interne), tandis que les ventes de licences sur site par les éditeurs de logiciels traditionnels relevaient de la norme 985-20. Cette distinction pouvait être contre-intuitive sur le plan économique — deux bases de code fonctionnellement similaires pouvaient être traitées différemment simplement en raison du modèle de livraison (Source: [schneiderdowns.com](https://www.schneiderdowns.com)) (Source: www.crowe.com). Par exemple, Schneider Downs observe qu'une entreprise qui vendait autrefois des licences de son logiciel et qui est passée par la suite à un modèle d'abonnement SaaS comptabilisera désormais une grande partie de sa R&D comme une charge d'usage interne (Source: [schneiderdowns.com](https://www.schneiderdowns.com)) (Source: www.crowe.com).

Alors que la comptabilité des logiciels à usage interne prenait du retard par rapport à l'évolution des pratiques, les incohérences se sont multipliées. De multiples analyses et enquêtes ont documenté cet effet. Une enquête d'Armanino auprès des 100 plus grandes entreprises SaaS publiques a révélé que **69 % des entreprises SaaS comptabilisent les coûts de développement en immobilisations** dans la pratique (contre 62 % en 2017) (Source: www.armanino.com). Pendant ce temps, Wall Street Prep souligne que des entreprises comme Alphabet Inc. (Google) comptabilisent *effectivement* leurs coûts de logiciels internes après la planification préliminaire, alors que d'autres ne le font pas, ce qui entraîne d'importantes différences dans les états financiers (Source: www.wallstreetprep.com) (Source: www.wallstreetprep.com). Finalement, le FASB a pris des mesures pour moderniser la directive : en septembre 2025, il a publié l'ASU 2025-06, officiellement intitulée « *Améliorations ciblées de la comptabilisation des logiciels destinés à un usage interne* » (Source: rsmus.com) (Source: dart.deloitte.com). Cette mise à jour, décrite comme une « *amélioration ciblée* », remplace les étapes rigides par un seuil fondé sur des principes, conçu pour fonctionner avec n'importe quelle méthodologie de développement.

Cadre historique des PCGR (ASC 350-40 vs ASC 985-20 vs ASC 350-50)

ASC 350-40 (Logiciels destinés à un usage interne). Selon la codification antérieure, l'ASC 350-40 exigeait que les entités traitent le développement de logiciels à usage interne par phases : (1) Projet préliminaire, (2) Développement d'applications et (3) Post-mise en œuvre/exploitation (Source: www.cohenco.com) (Source: www.wallstreetprep.com). Seuls les coûts engagés pendant la phase de développement des applications étaient comptabilisés en immobilisations ; les coûts des phases préliminaire ou de post-mise en œuvre étaient passés en charges (Source: www.cohenco.com) (Source: www.wallstreetprep.com). Pour entrer dans la phase de développement des applications, deux conditions préalables devaient être remplies : la phase de projet préliminaire devait être terminée **et** la direction devait avoir autorisé et engagé le financement (ce qui signifiait que le projet était susceptible d'être terminé) (Source: www.cohenco.com) (Source: www.wallstreetprep.com). Les activités préliminaires telles que la planification conceptuelle, l'évaluation des fournisseurs ou les évaluations technologiques étaient passées en charges au fur et à mesure qu'elles étaient engagées (Source: www.cohenco.com) (Source: www.wallstreetprep.com). Une fois la comptabilisation en immobilisations terminée (généralement lorsque le logiciel était substantiellement terminé et prêt à être utilisé), toute maintenance ultérieure ou mise à niveau de routine était passée en charges (Source: www.cohenco.com) (Source: www.wallstreetprep.com).

Il est important de noter que les règles de l'ASC 350-40 ne s'appliquent que lorsque ses *deux* critères sont remplis (engagement de la direction et probabilité d'achèvement). Les coûts comptabilisés en immobilisations sont ceux « pour développer ou obtenir des logiciels destinés à un usage interne » tels qu'énumérés dans l'ASC 350-40-25-1 à 25-5, y compris les *coûts directs externes* (par exemple, les développeurs tiers) et la *masse salariale interne* pour le personnel affecté au projet (Source: www.scribd.com) (Source: www.wallstreetprep.com). Les frais généraux ou les coûts « G&A » (comme le soutien à la gestion générale, la formation ou la maintenance) sont explicitement exclus de la comptabilisation en immobilisations (Source: www.scribd.com) (Source: www.cohenco.com). Après l'ASU 2025-06, bien que le **moment** de la comptabilisation soit modifié, les types de coûts admissibles à une comptabilisation en immobilisations restent largement les mêmes (la nouvelle ASU ne modifie pas le champ d'application des coûts comptabilisables) (Source: www.eidebailly.com) (Source: www.cohenco.com).

ASC 350-50 (Développement de sites Web). Avant l'ASU 2025-06, les coûts de développement de sites Web et de certaines applications relevaient de l'ASC 350-50. Ce sous-thème avait des règles similaires : les coûts de la phase préliminaire étaient passés en charges, et les coûts de développement des applications pouvaient être comptabilisés en immobilisations s'ils étaient utilisés en interne (ou à des fins publicitaires, etc., sous des conditions strictes) (Source: www.wallstreetprep.com). La nouvelle ASU **remplace explicitement l'ASC 350-50**, intégrant le développement de sites Web dans le cadre plus large de la norme 350-40 (Source: rsmus.com) (Source: www.bakertilly.com). En d'autres termes, le même seuil de « probabilité d'achèvement » s'applique désormais aux sites Web comme aux applications utilisées en interne, éliminant ainsi un ensemble de règles distinctes.

ASC 985-20 (Logiciels destinés à la vente, à la location ou à la commercialisation). Cette sous-rubrique régit les logiciels destinés à des tiers (par exemple, les produits logiciels ou les applications sous licence). En vertu de l'ASC 985-20, les entreprises ne capitalisent les coûts qu'une fois la *faisabilité technologique* atteinte (souvent définie par l'achèvement d'une conception détaillée ou d'un premier modèle fonctionnel) et avant la mise sur le marché du produit (Source: www.wallstreetprep.com) (Source: www.armanino.com). L'IAS 985-20 utilise des termes tels que *faisabilité technologique* et diffère la comptabilisation des coûts jusqu'à la sortie du produit. À l'inverse, les règles relatives à l'usage interne n'utilisaient *pas* le concept de faisabilité technologique : elles se concentraient plutôt sur les étapes du projet et la probabilité d'achèvement. Il est à noter que l'ASU 2025-06 **ne modifie pas** l'ASC 985-20. Les deux cadres divergent toujours : un développement logiciel fonctionnellement identique pourrait être comptabilisé différemment selon que l'entreprise prévoit de le vendre/concéder sous licence ou simplement de l'utiliser en interne (Source: schneiderdowns.com) (Source: www.crowe.com). (Un exemple clé : si un abonné SaaS ne peut pas prendre possession d'une copie complète du logiciel, l'accord est traité comme un service, ce qui signifie que les coûts de développement restent des coûts d'usage interne selon l'ASC 350-40 (Source: www.crowe.com), même si le résultat est un produit cloud générant des revenus.)

Contexte SaaS. En pratique, les **fournisseurs SaaS** relèvent presque toujours de l'ASC 350-40 pour leurs coûts de développement. Schneider Downs souligne : « Les entités Software-as-a-Service (SaaS) relèvent généralement des directives de l'ASC 350-40. Si le client n'a pas le droit de prendre possession du logiciel, le contrat est un contrat de service, et non une licence » (Source: [schneiderdowns.com](https://www.schneiderdowns.com)). Ainsi, la plateforme d'une entreprise SaaS est traitée comme un actif interne, bien qu'il s'agisse de son moteur de revenus principal. Pour cette raison, les entreprises SaaS ont eu tendance à capitaliser une part importante de leur R&D (par exemple, les nouvelles fonctionnalités ou les mises à niveau de la plateforme) en vertu de l'ASC 350-40, en les amortissant sur leur durée d'utilité (souvent 2 à 5 ans) (Source: www.wallstreetprep.com) (Source: www.cohenco.com). À l'inverse, les entreprises de logiciels « en boîte » traditionnels capitalisent plus tard selon l'ASC 985-20 (après la faisabilité technique) et cessent avant la mise sur le marché. Cette divergence a incité certains à plaider pour un « modèle unique » traitant tout développement logiciel de la même manière. En effet, des praticiens éminents, comme l'ancien directeur financier de Salesforce, Andrew Hyde, soutiennent que « les coûts de développement de logiciels devraient être capitalisés de manière cohérente, qu'ils soient internes ou externes », notant que les deux méthodes peuvent produire pratiquement le même résultat grâce au jugement de la direction, et que l'approche duale peut donc semer la confusion au sein des équipes comptables et de développement (Source: www.armanino.com).

Limites des directives antérieures

Les préparateurs et les auditeurs ont tous deux souligné des **problèmes d'opérabilité** dans l'ASC 350-40. Selon l'ancien modèle, la capitalisation se déclenchait *automatiquement* dès l'entrée dans la phase de développement de l'application, mais dans de nombreux scénarios Agile, il n'existe pas de point d'entrée clair. Les entreprises ont souvent dû interpréter ce qui constituait la « sélection finale » de la conception ou le moment où la phase préliminaire se terminait réellement (Source: www.rit.edu) (Source: www.wallstreetprep.com). En pratique, cela a conduit à une diversité de traitements : certaines entreprises ont commencé à capitaliser lorsqu'un plan de projet détaillé a été approuvé (ou qu'un contrat de fournisseur a été signé), tandis que d'autres ont attendu que le codage soit substantiellement engagé. Comme le note Baker Tilly, les parties prenantes ont exprimé que les directives héritées « sont obsolètes et manquent de pertinence compte tenu de l'évolution du développement logiciel — en particulier, la transition d'une méthode prescriptive et séquentielle... vers une méthode incrémentale et itérative » (Source: www.bakertilly.com). Ce décalage a entraîné des résultats incohérents : deux modules logiciels développés en parallèle pouvaient être capitalisés selon des calendriers différents en fonction de détails définitionnels.

Par exemple, le *projet X* pouvait être en phase de collecte continue des besoins, ce qui rendait difficile de savoir s'il était encore au stade « préliminaire » ou « de développement » à un moment donné. Ou bien, le *projet Y* pouvait avoir bénéficié de nouvelles fonctionnalités ajoutées après la mise en service initiale, soulevant des questions sur la nécessité de capitaliser ou de passer en charges cette partie selon les règles de maintenance (Source: www.cohenco.com) (Source: www.wallstreetprep.com). De nombreux praticiens (y compris des directeurs financiers de sociétés SaaS cotées) ont plaidé pour un déclencheur davantage fondé sur des principes. Ces préoccupations ont finalement conduit le FASB à proposer et à adopter l'ASU 2025-06, qui recentre la décision sur l'autorisation de la direction et le risque du projet, plutôt que sur l'achèvement d'une étape procédurale (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.armanino.com).

Changements clés dans l'ASU 2025-06 (ASC 350-40)

En septembre 2025, le FASB a publié l'**ASU 2025-06, Améliorations ciblées de la comptabilisation des logiciels destinés à un usage interne**, applicable aux exercices ouverts après le 15 décembre 2027 (avec possibilité d'adoption anticipée) (Source: [rsmus.com](https://www.rsmus.com)) (Source: [dart.deloitte.com](https://www.dart.deloitte.com)). Cette mise à jour modernise l'ASC 350-40 de plusieurs manières importantes :

- **Élimination des étapes de projet** : L'ASU 2025-06 **supprime toutes les références aux étapes préliminaires, de développement d'application et post-implémentation** dans l'ASC 350-40 (Source: www.bakertilly.com) (Source: www.bakertilly.com). Au lieu de cela, il existe un point de décision unique basé uniquement sur des facteurs économiques, et non sur la position du projet dans un cycle de vie nominal.
- **Seuil de probabilité d'achèvement** : La capitalisation ne commence *que lorsque deux critères sont remplis simultanément* (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com) : (1) l'**autorisation et l'engagement de financement** ont eu lieu, et (2) il est **probable** (selon la définition des PCGR) que le projet soit achevé et que le logiciel soit utilisé comme prévu (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com). L'exigence de « probabilité » utilise le sens du glossaire principal de « susceptible de se produire ». L'ASU 2025-06 souligne que les deux conditions doivent être remplies pour commencer la capitalisation (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com). Cela dépend du jugement de la direction : par exemple, l'exécution d'un contrat de développement ou l'approbation d'un budget constitue une autorisation, tandis que l'évaluation des incertitudes techniques/de conception relève de la probabilité (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com).

- **Incertitude de développement significative (Nouveau verrou)** : Il est crucial de noter que si une **incertitude de développement significative** existe *au moment de l'évaluation*, le seuil est considéré comme **non atteint** (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). L'ASU 2025-06 définit explicitement deux facteurs indiquant une « incertitude de développement significative » (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com) : (a) le logiciel présente des fonctionnalités ou une technologie nouvelles/non éprouvées dont le risque n'a pas été résolu par des tests, et (b) les exigences de performance significatives n'ont pas été identifiées ou sont encore sujettes à des révisions substantielles (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com). Si l'un de ces facteurs est présent, la capitalisation doit être **différée** jusqu'à ce que l'incertitude soit levée. Par exemple, si une entreprise développe un moteur d'analyse IA de pointe qui n'a jamais été construit auparavant, les obstacles techniques inconnus retarderaient la satisfaction de la condition de « probabilité » (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com).
- **Alignement pour tous les modèles de livraison** : L'ASU s'applique **de manière égale aux projets de logiciels à usage interne sur site, hébergés ou basés sur le cloud**, ce qui le rend agnostique vis-à-vis de la technologie (Source: www.crowe.com). Il **remplace** également formellement l'**ASC 350-50** (Coûts de développement de sites Web) et intègre le développement Web/applications dans la norme 350-40 (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Ainsi, une entreprise qui construit un portail Web interne et une autre qui met à jour une application mobile d'entreprise utilisent le même test de capitalisation. Ce modèle unifié couvre les projets informatiques traditionnels, les implémentations ERP, le développement de plateformes SaaS et les sites Web sous une seule rubrique ASC.
- **Modifications des informations à fournir** : La mise à jour précise que les **informations à fournir pour les logiciels à usage interne capitalisés** suivent les règles de l'ASC 360 (Immobilisations corporelles), et non les informations générales sur les actifs incorporels (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com). Cela signifie que les logiciels capitalisés doivent être traités comme des immobilisations dans les tableaux de flux de trésorerie et les notes annexes (par exemple, présentés dans les flux de trésorerie liés aux investissements) (Source: www.armanino.com) (Source: www.eidebailly.com). Avant l'ASU 2025-06, de nombreuses entreprises présentaient les logiciels à usage interne soit sous les actifs incorporels (ASC 350), soit sous les immobilisations corporelles ; la nouvelle directive normalise cette présentation.
- **Dispositions de transition** : Les entités peuvent adopter la norme de manière prospective, prospective modifiée ou rétrospective (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Une approche prospective applique le nouveau test uniquement aux coûts engagés après l'adoption. Avec une méthode modifiée, les projets en cours qui *étaient* capitalisables selon les anciennes règles mais qui *ne l'auraient pas été* selon les nouvelles règles sont passés en charges via un ajustement ponctuel. Dans le cadre d'une approche rétrospective complète, les périodes antérieures sont ajustées. Le FASB autorise explicitement l'adoption anticipée (au début de n'importe quel exercice annuel) (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com), afin de permettre aux entreprises d'aligner leur comptabilité sur leurs pratiques de développement réelles plus rapidement si elles le souhaitent.

En résumé, l'**ASU 2025-06 recentre la décision de capitalisation sur la prudence plutôt que sur le processus**. Comme l'explique Crowe LLP, la nouvelle directive ne tente *pas* de modifier les coûts capitalisables ; elle **reformule uniquement le moment où la capitalisation commence** (Source: www.crowe.com) (Source: [rsmus.com](https://www.rsmus.com)). Tous les analystes soulignent que c'était l'intention du FASB : « moderniser les directives... rendre les règles neutres vis-à-vis de la méthodologie de développement et plus opérationnelles pour les environnements itératifs » (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Pour les logiciels SaaS et basés sur le cloud, la mise à jour devrait avoir **l'impact le plus important**, car ces entreprises capitalisaient historiquement une part importante de leur R&D en tant qu'usage interne (Source: www.crowe.com) (Source: [rsmus.com](https://www.rsmus.com)). À l'avenir, les jugements comptables clés se concentreront sur la *définition de la portée du projet logiciel*, l'*identification des exigences de performance significatives* et la *documentation précise du moment où l'incertitude de développement est levée* (Source: www.crowe.com) (Source: www.crowe.com). Les exemples détaillés dans la section des fondements de l'ASU (illustrés ci-dessous) soulignent cette approche.

Impact sur les entreprises de logiciels SaaS et Cloud

L'ASU 2025-06 a des implications spécifiques pour les **développeurs SaaS** et les entreprises de logiciels basés sur le cloud, dont les modèles commerciaux brouillent souvent les frontières entre usage interne et produit livré. Selon les anciens PCGR, de nombreux fournisseurs SaaS avaient pour pratique de capitaliser une grande partie des coûts de développement en vertu de l'ASC 350-40, même si ces coûts construisaient effectivement la plateforme destinée aux clients. À l'inverse, les entreprises de logiciels sous licence traditionnels passaient en grande partie la R&D en charges jusqu'à la faisabilité technologique. L'ASU 2025-06 ne *force pas* les entreprises SaaS à passer à l'ASC 985-20 – chaque accord est toujours évalué selon le test de licence existant. Cependant, la nouvelle directive facilite la capitalisation des coûts par les entreprises SaaS dans le cadre d'un processus de développement agile.

Pour les développeurs SaaS, **le jugement est désormais différent**. L'accent est mis sur le moment où la direction s'est *engagée, implicitement ou explicitement*, dans un projet et sur la probabilité de sa réussite (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.eidebailly.com). Les entreprises SaaS doivent identifier chaque initiative de développement entrant dans le champ d'application de l'ASC 350-40 (absence de projet de commercialisation externe), puis appliquer les critères de l'ASU. Pour chaque projet, la direction doit évaluer en permanence si une incertitude significative subsiste. Armanino

LLC rapporte que près de **69 % des entreprises SaaS cotées en bourse capitalisent actuellement leurs coûts de développement** (Source: www.armanino.com), ce qui suggère que de nombreux projets répondent déjà aux critères traditionnels. Après l'ASU, ces mêmes projets pourraient déclencher une capitalisation **plus tôt** (puisque la satisfaction du test ne dépend plus de l'achèvement d'une phase de planification formelle), mais pourraient également être **différés plus longtemps** si l'incertitude persiste.

Le jugement dans le développement Agile

Une caractéristique du développement SaaS est la publication itérative et l'amélioration continue. En vertu de l'ASU 2025-06, les équipes agiles doivent documenter avec soin le moment où les seuils clés sont atteints. Par exemple, supposons que l'équipe produit d'une entreprise SaaS commence à créer une nouvelle fonctionnalité lors d'un sprint Agile. La direction a approuvé le budget du projet (autorisation) et estime que la fonctionnalité apportera de la valeur au client (probable). Cependant, tant que l'équipe n'a pas verrouillé le périmètre final et résolu les éventuels problèmes techniques, la condition de « probabilité d'achèvement » peut ne pas encore être satisfaite (Source: www.eidebailly.com). Dans ce scénario, aucun coût ne serait capitalisé initialement ; tous les sprints techniques restent comptabilisés en charges. Une fois le périmètre stabilisé et toutes les incertitudes majeures (par exemple, un algorithme inédit) levées, l'entreprise peut commencer à capitaliser les coûts de manière rétroactive (Source: www.eidebailly.com) (Source: www.eidebailly.com).

À l'inverse, considérons une entreprise SaaS qui engage un développeur tiers pour créer un module entièrement spécifié. Lors de la signature du contrat (et de l'approbation des fonds), la direction a à la fois autorisé le projet et conclu que son achèvement est probable (peut-être compte tenu des antécédents du fournisseur). Si le module n'implique aucune innovation « non éprouvée », le seuil de probabilité est atteint immédiatement. Dans ce cas, tous les salaires, honoraires de fournisseurs et coûts connexes peuvent être capitalisés à partir de ce moment (Source: www.eidebailly.com). Cela reflète l'**Exemple 1** des illustrations de l'ASU : une société de services exécutant un contrat de développement au 1er mai commence à capitaliser ce jour-là (Source: www.eidebailly.com). De tels résultats soulignent que pour les entreprises SaaS, les **critères sont désormais prospectifs et dépendent des faits plutôt que du calendrier**.

Il est important de noter que l'ASU 2025-06 ne **modifie pas les coûts éligibles**. Les développeurs SaaS ne doivent toujours capitaliser que les coûts de développement étroitement liés à la création ou à la configuration de l'application (Source: www.wallstreetprep.com) (Source: www.cohenco.com). La maintenance ordinaire des produits, la formation des utilisateurs ou le support général continueront d'être comptabilisés en charges (Source: www.cohenco.com) (Source: www.wallstreetprep.com). Par exemple, une entreprise SaaS peut engager des dépenses de formation du personnel après la sortie d'une nouvelle plateforme ; ces coûts de formation post-implémentation sont spécifiquement comptabilisés en charges (Source: www.cohenco.com) (Source: www.wallstreetprep.com). De même, les améliorations qui corrigent simplement des bugs sans ajouter de fonctionnalités restent comptabilisées en charges. En pratique, de nombreuses entreprises mapperont ces règles à leur comptabilité analytique : les coûts sous des catégories de tâches telles que « Codage » ou « Configuration » sont capitalisables, tandis que les « Tests » et la « Formation des utilisateurs » pourraient nécessiter une analyse (les tests sont généralement capitalisables, **sauf** pour les corrections de bugs de routine) (Source: www.wallstreetprep.com).

Comparabilité et effets sur le reporting

Étant donné que l'ASU 2025-06 entraîne souvent une **capitalisation plus précoce** dans les projets itératifs, elle aura tendance à **gonfler les actifs et à réduire les charges aux premiers stades** du développement (par rapport à l'ancien modèle). À l'inverse, les projets entravés par l'incertitude montreront une capitalisation *plus tardive*. L'effet net sur les états financiers d'une entreprise dépend du nombre de projets franchissant le seuil. Cependant, il existe un large consensus sur le fait que ce changement **améliore la comparabilité**. Comme le note un livre blanc d'Armanino, selon les PCGR actuels, des projets similaires peuvent présenter des « états financiers différents » selon le phasage ; la mise à jour vise à ce que la comptabilité reflète la substance économique (actifs de R&D à long terme) indépendamment du contrat ou de la méthodologie (Source: www.wallstreetprep.com) (Source: www.armanino.com). En fait, l'enquête d'Armanino cite un ancien directeur financier SaaS qui considère ces améliorations ciblées comme un pas vers « l'alignement sur les pratiques des entreprises SaaS, en s'éloignant de la norme de "faisabilité technique" », et préconise une unification à terme de la comptabilité des logiciels internes et externes (Source: www.armanino.com) (Source: www.armanino.com).

Du point de vue de l'information financière, les entreprises de logiciels présenteront désormais tous leurs développements capitalisés sous la rubrique Immobilisations corporelles (PP&E). L'ASU 2025-06 exige que ces coûts de logiciels différés soient présentés comme une sortie de trésorerie liée à l'investissement dans le tableau des flux de trésorerie (comme toute autre dépense d'investissement) (Source: www.armanino.com). Les entités doivent divulguer la nature des coûts de logiciels capitalisés et leur période d'amortissement, mais elles n'ont plus besoin de faire des divulgations distinctes sur les actifs incorporels (Source: rsmus.com) (Source: www.eidebailly.com). En pratique, on peut s'attendre à des changements dans les notes de bas de page (par exemple, les notes sur les immobilisations corporelles incluront les logiciels dans les tableaux de coûts).

Perspective de la SEC et réglementaire

À ce jour, les dépôts auprès de la SEC suggèrent que la Commission s'est davantage concentrée sur la reconnaissance des revenus et les problèmes de capitalisation des commissions que sur les logiciels à usage interne. Le rapport d'Armanino n'a trouvé **aucune lettre de commentaires de la SEC** sur la diversité des dépenses de développement de logiciels au cours des deux dernières années (Source: www.armanino.com), même si les lettres de commentaires sur l'ASC 340-40 (coûts des commissions) ont augmenté (Source: www.armanino.com). Cela peut refléter le fait que la distinction entre l'ASC 985-20 et l'ASC 350-40 est une pratique établie de longue date pour la plupart des entreprises publiques, et que les changements apportés par la nouvelle ASU ne créent pas de discontinuité de nature à alarmer les régulateurs. Néanmoins, les entreprises qui se préparent à l'adoption doivent être conscientes que le traitement des flux de trésorerie par l'ASU (exigeant que les coûts des logiciels internes soient des sorties de trésorerie liées à l'investissement) pourrait affecter les divulgations et les indicateurs de flux de trésorerie.

Dans l'ensemble, pour les développeurs SaaS, le **changement fondamental** est opérationnel : élaborer des politiques cohérentes pour évaluer et documenter le nouveau seuil de capitalisation pour chaque effort de développement majeur. Des équipes mixtes composées de membres de la finance et de l'informatique devraient collaborer pour définir ce qui constitue une unité de « projet logiciel », identifier rapidement les exigences de performance significatives et suivre le moment où les incertitudes sont résolues (Source: www.eidebailly.com) (Source: www.crowe.com). NetSuite, en tant que système d'entreprise, peut être configuré pour soutenir ce processus (voir ci-dessous). D'un point de vue stratégique, de nombreux directeurs financiers SaaS considèrent l'ASU 2025-06 comme la codification de la réalité selon laquelle le code développé en interne est un actif, alignant la comptabilité sur la façon dont les autres activités de R&D sont traitées (par exemple, la capitalisation selon l'article 174 pour la fiscalité (Source: www.armanino.com). Une entreprise SaaS tournée vers l'avenir devrait évaluer ses budgets, ses flux de trésorerie et sa planification fiscale à la lumière de la R&D capitalisée qui pourrait en résulter.

Caractérisation des coûts capitalisables par rapport aux coûts comptabilisés en charges

Selon l'ancienne et la nouvelle directive, seuls certains **coûts de développement directs** peuvent être capitalisés ; les autres doivent être comptabilisés en charges au cours de la période où ils sont engagés. L'ASU 2025-06 n'élargit ni ne réduit la liste des coûts éligibles, mais la compréhension de cette liste est essentielle à la mise en œuvre. Selon l'ASC 350-40, les coûts des logiciels à usage interne capitalisables comprennent généralement : (a) les **coûts directs externes** de matériaux et de services (payés à des fournisseurs ou consultants tiers) spécifiquement pour le projet logiciel, et (b) les **coûts de masse salariale interne et les coûts liés à la masse salariale** pour les employés qui sont directement associés au projet et qui y consacrent du temps (Source: www.scribd.com). Par exemple, une entreprise peut engager des consultants pour coder un CRM personnalisé et affecter en interne une équipe de développeurs ; les honoraires des consultants et les salaires des développeurs (et leurs avantages sociaux) peuvent être capitalisés une fois les critères remplis (Source: www.rit.edu) (Source: www.scribd.com). Il est important de noter que la rémunération à base d'actions pour les employés travaillant sur le projet est également incluse dans les coûts de masse salariale capitalisables.

Les frais généraux indirects ou les coûts administratifs et généraux (y compris le temps consacré par les cadres, le marketing, le loyer, les services publics ou le support informatique général) ne sont **pas** capitalisés (Source: www.scribd.com) (Source: www.wallstreetprep.com). Par exemple, le temps que le PDG consacre aux réunions de supervision n'est pas éligible. Les coûts de formation sont généralement comptabilisés en charges ; l'ASC 350-40 exclut la capitalisation de la formation des employés, même pendant le développement de l'application (Source: www.cohenco.com) (Source: www.wallstreetprep.com). De même, les coûts tels que les voyages ou l'hébergement ne sont capitalisables *que s'ils* sont directement imputables au projet (par exemple, des consultants voyageant spécifiquement pour implémenter le logiciel) (Source: www.rit.edu). Une règle empirique utile : les coûts qui n'auraient pas été engagés « sans » le projet de développement doivent être capitalisés ; tout ce qui aurait été une dépense d'exploitation normale reste une charge.

Une fois capitalisé, l'actif est amorti sur sa durée d'utilité. L'ASC 350-40 exige depuis longtemps un amortissement linéaire (sans mention de dépréciation, sauf si le projet est abandonné) (Source: www.wallstreetprep.com). En pratique, les entreprises SaaS amortissent généralement sur 2 à 5 ans selon l'obsolescence technologique attendue. Par exemple, AthenaHealth a amorti sa plateforme interne sur 2 à 5 ans (Source: www.wallstreetprep.com) ; Alphabet (Google) a également indiqué que les durées de vie de ses logiciels à usage interne se situent généralement dans cette fourchette (Source: www.wallstreetprep.com). Les durées de vie effectives sont une question de politique et d'estimation ; les entreprises doivent justifier que les coûts capitalisés sont mis en adéquation avec les revenus ou les avantages qu'ils permettent.

Perspectives et opinions

De multiples perspectives sectorielles et d'experts aident à éclairer l'importance de l'ASU 2025-06 :

- **FASB/Normalisateurs** : La base des conclusions du FASB (et le communiqué de presse qui l'accompagne) souligne que le nouveau modèle préserve les critères de capitalisation existants mais « aligne mieux la comptabilité... avec les approches de développement actuelles » en supprimant le « cloisonnement par étapes » prescriptif (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Un « Heads Up » de Deloitte note de même que le FASB a examiné une invitation à commenter (ITC) qui soulignait le besoin de modernisation. Il est important de noter que le FASB a décidé de *ne pas* aligner la comptabilité des logiciels internes sur l'ASC 985-20, ce qui signifie que les logiciels internes et externes suivent toujours des cadres distincts (Source: [dart.deloitte.com](https://www.dart.deloitte.com)). Le Conseil considère l'ASU 2025-06 comme un changement ciblé, et non comme une refonte complète.
- **Cabinets comptables et analystes** : Les grands cabinets ont publié des guides sur l'ASU 2025-06. RSM explique le nouveau concept de « seuil de reconnaissance de probabilité d'achèvement » et fournit des tests pratiques ; Baker Tilly souligne la suppression de toutes les références aux étapes du projet (Source: [rsmus.com](https://www.rsmus.com)) (Source: www.bakertilly.com). Crowe souligne que la version du test de probabilité d'achèvement exige désormais explicitement que les entreprises **recherchent toute incertitude significative** dans chaque projet, et que la documentation sera essentielle (Source: www.crowe.com) (Source: www.crowe.com). Ces analyses trouvent un écho auprès des praticiens : le nouveau modèle déplace les décisions de jugement des cloisonnements basés sur des étiquettes vers une évaluation fondée sur des preuves.
- **Point de vue de la finance d'entreprise (CFO/Contrôleur)** : Du côté des entreprises, la mise à jour a été largement perçue comme une clarification. Dans un livre blanc d'enquête d'Armanino, l'**ancien directeur financier SaaS Andrew Hyde** a observé que les amendements « semblent avoir été dérivés de la prise en compte de l'approche par modèle unique » et « reflètent ce qui se passe déjà alors que les entreprises de logiciels s'alignent sur les pratiques des entreprises SaaS ». Hyde critique néanmoins la divergence continue des méthodes, arguant qu'en fin de compte, la capitalisation des logiciels internes et externes conduit de toute façon à des résultats similaires (Source: www.armanino.com). Dans l'ensemble, de nombreux dirigeants financiers SaaS accueillent favorablement cette flexibilité : moins de contraintes artificielles signifie aligner la comptabilité sur la livraison agile. Cependant, ils préviennent que des livrables d'ingénierie importants (comme un module d'IA propriétaire) peuvent encore nécessiter de longues phases de R&D avec des résultats intrinsèquement incertains.
- **Investisseurs et analystes** : Bien que les analystes actions commentent rarement explicitement la capitalisation des logiciels, cette décision a des implications pour les indicateurs clés. En permettant une capitalisation plus précoce, l'ASU 2025-06 **diffère effectivement les dépenses vers des périodes ultérieures**, augmentant le résultat opérationnel à court terme. Cela pourrait modestement améliorer les profils d'EBITDA et de marge pour les entreprises de logiciels à forte croissance en transition. À l'inverse, les projets encore en cours de flux porteront plus de charges dès le départ, rendant potentiellement les pertes actuelles plus importantes mais plus claires. Les analystes surveilleront si l'augmentation des soldes d'actifs incorporels (et de l'amortissement) modifie les modèles de valorisation. En somme, ce changement pourrait rendre les bénéfices déclarés des entreprises SaaS plus comparables, réduisant une source de diversité entre les modèles économiques sur site et dans le cloud.
- **Fiscalité et réglementation** : Les autorités fiscales ont également changé la donne récemment. Plus notable encore, l'article 174 de l'IRC a été modifié (avec une entrée en vigueur en 2022) pour **exiger que les dépenses de R&D – y compris le développement de logiciels internes – soient capitalisées et amorties sur une période de 5 à 15 ans** (Source: www.armanino.com). De nombreuses entreprises SaaS se sont retrouvées à payer des impôts bien plus élevés, car ce qui était auparavant des charges déductibles doit désormais être amorti. Armanino rapporte des cas d'entreprises confrontées à une charge fiscale « trois à quatre fois » supérieure à celle de l'année précédente en raison de ce changement (Source: www.armanino.com). L'ASU 2025-06 est **indépendante du droit fiscal** (elle ne modifie pas l'article 174), mais les deux tendances se renforcent mutuellement : les activités capitalisées selon les PCGR (GAAP) le seront généralement aussi sur le plan fiscal, ce qui augmentera les soldes d'impôts différés. Les entreprises doivent anticiper ces effets sur les flux de trésorerie. Sur le plan réglementaire, la division des finances des entreprises de la SEC n'a signalé aucune nouvelle préoccupation spécifique concernant la capitalisation des logiciels internes (aucune lettre de commentaire n'a été émise à ce sujet récemment (Source: www.armanino.com), mais les émetteurs doivent s'assurer que leurs informations financières expliquent clairement tout changement dans la politique de capitalisation et la manière dont les actifs incorporels ajoutés sont amortis.

Guide de mise en œuvre pour les développeurs SaaS (Focus NetSuite)

Pour les entreprises SaaS utilisant NetSuite comme système ERP/financier, la conformité à l'ASU 2025-06 impliquera la configuration de NetSuite pour suivre les dépenses de projet et allouer les coûts entre immobilisations et charges. Une configuration appropriée garantit qu'une fois qu'un projet dépasse le seuil de capitalisation, tous les coûts éligibles sont comptabilisés dans un **compte d'actif immobilisé**, destiné à être amorti, plutôt que d'être passés en charges courantes. Voici des étapes et fonctionnalités pratiques que les équipes financières SaaS devraient envisager lors de l'utilisation de NetSuite sous les nouvelles règles :

1. Définir les projets logiciels et les budgets : Créez des **enregistrements de projet** distincts (en utilisant le module Projets ou Job Costing de NetSuite) pour chaque initiative de développement significative. Assignez un *chef de projet* et des tâches (ex. : conception, codage, tests). En utilisant la budgétisation de projet NetSuite, allouez les heures et les coûts prévisionnels par phase et par type de ressource (Source: docs.oracle.com). Cela permet à la direction de surveiller les dépenses en temps réel pendant le développement. Il est crucial que les projets identifiés comme éligibles à la capitalisation soient signalés (par exemple, via un champ personnalisé de projet) afin que les coûts puissent être isolés. Par exemple, le RIT suggère d'utiliser une catégorie spécifique (comme un code FEC ou une classification de projet NetSuite) pour les coûts de mise en œuvre SaaS capitalisables (Source: www.rit.edu).

2. Configurer le plan comptable / les types de dépenses : Grâce à la fonctionnalité **Job Costing** de NetSuite, vous pouvez définir des *Types de dépenses de projet* qui déterminent les comptes du grand livre vers lesquels les saisies de temps et de dépenses sont comptabilisées (Source: docs.oracle.com). Nous recommandons de créer des catégories de dépenses de projet distinctes pour le « Développement en capital », les « Dépenses de R&D » et le « Support ». Par exemple, un type de dépense « Développement en capital » pourrait pointer vers un compte d'actif non courant **Développement de logiciel différé**, tandis que les types de dépenses « Maintenance » pointeraient vers des comptes de charges de résultat. Lorsque les membres de l'équipe saisissent du temps/des coûts sur le projet, le type de dépense de projet créditera le compte approprié. Dans les configurations OneWorld, des ajustements inter-sociétés peuvent également être générés automatiquement si la R&D est incubée dans une filiale mais passée en charges dans une autre (Source: docs.oracle.com).

3. Enregistrement des transactions capitalisables : Une fois qu'un projet atteint le seuil de capitalisation de l'ASU, toutes les dépenses éligibles doivent être codées vers les comptes d'*actif* désignés. Dans NetSuite, cela peut signifier l'utilisation de **catégories de dépenses** ou d'**enregistrements d'articles** liés aux comptes de capital. Par exemple, les coûts de main-d'œuvre interne (capturés via les feuilles de temps) seraient dirigés vers le compte d'actif R&D différé (via le Job Costing) au lieu d'un compte de charges salariales. De même, les factures des fournisseurs pour les développeurs sous contrat seraient codées avec une catégorie de dépenses qui s'impute au même compte d'actif. (**L'activation de l'amortissement** de NetSuite peut être utilisée ici : en assignant un modèle d'amortissement à la ligne de dépense, le système créera automatiquement un report.) L'objectif est que toutes les dépenses capitalisables s'accumulent dans un compte de bilan plutôt que dans le compte de résultat pendant la période de capitalisation.

4. Amortissement et suivi des actifs : Une fois les coûts capitalisés, ils peuvent être **amortis** sur leur durée d'utilité. NetSuite propose deux approches principales :

- **Modèles d'amortissement :** Activez la fonctionnalité **Amortissement** (Enable > Feature > Accounting > Amortization). Créez un modèle d'amortissement pour les coûts de développement logiciel en précisant les périodes de durée d'utilité (par exemple, 36 mois). Définissez la méthode de comptabilisation des charges (ex. : linéaire) et liez-la aux comptes d'actif différé (Source: netsuitedocumentation1.gitlab.io) (Source: netsuitedocumentation1.gitlab.io). Lorsque les transactions pertinentes sont enregistrées (factures, notes de frais avec ce modèle), NetSuite générera des écritures de journal planifiées à chaque période, déplaçant une partie du solde différé vers les charges d'amortissement. Cela s'aligne sur l'exigence du FASB selon laquelle les logiciels capitalisés doivent être amortis comme un actif à long terme. Concrètement, un coût de projet de 120 000 \$ pourrait être amorti à hauteur de 4 000 \$ par mois sur 30 mois, par exemple.
- **Module d'immobilisations :** Alternativement, si l'entreprise dispose de la SuiteApp **Fixed Assets Management** de NetSuite, on peut traiter le logiciel capitalisé comme un actif incorporel. Dans cette option, créez un enregistrement d'actif pour le logiciel terminé (avec le coût total capitalisé et la durée d'utilité). Le système automatisera alors les écritures de dépréciation/charge. Cette méthode peut rationaliser le reporting (le logiciel apparaît dans le registre des actifs) et assure l'intégration avec une comptabilité multi-référentiels si nécessaire.

5. Reporting et suivi : Utilisez les rapports intégrés de NetSuite pour vérifier les soldes. Le rapport **Rentabilité du projet** (avec l'onglet Coûts réels) peut être personnalisé pour afficher le capital par rapport aux charges par projet (Source: docs.oracle.com). Les rapports de **Balance de vérification** ou de **Bilan** de NetSuite devraient désormais inclure une ligne pour les Actifs logiciels différés (regroupement de tous les projets capitalisés). À chaque clôture de période, réconciliez ce solde et assurez-vous que les écritures d'amortissement correctes ont eu lieu. La **Piste d'audit** de NetSuite (audit des transactions) peut appuyer la documentation : liez les écritures de journal aux identifiants de projet. De plus, les informations périodiques (en notes de bas de page) devraient détailler le total des logiciels capitalisés et l'amortissement (net des cessions), ce que le reporting NetSuite peut fournir à partir de ces comptes.

6. Gouvernance et changements de processus : Au-delà de la configuration du système, les entreprises doivent adapter leurs processus. Les équipes de projet doivent déclencher des notifications au service financier lorsqu'un projet dépasse le seuil de l'ASU. Les cadres de contrôle (ex. : SOX) doivent inclure l'examen des chartes de projet par rapport aux coûts réels. NetSuite peut faciliter les approbations (ex. : flux de travail d'approbation sur l'achèvement d'une tâche de projet marquant « prêt à capitaliser »). Toutes les preuves (procès-verbaux de réunion, validations techniques, résultats de tests) justifiant la « probabilité d'achèvement » doivent être documentées comme support d'audit, bien que cela puisse résider en dehors de NetSuite.

7. Exemple de configuration : Pour illustrer concrètement, considérons une entreprise SaaS mettant en œuvre une nouvelle fonctionnalité. Elle configure le projet NetSuite « **Développement de la fonctionnalité X** », statut « En cours ». Le contrôleur configure deux types de dépenses : (a) *Main-d'œuvre Dev – Capital* mappant vers le compte d'actif R&D différé, et (b) *Main-d'œuvre Dev – Charge* vers un compte de résultat de R&D normal. Lorsque le responsable décide que le projet passe le test de l'ASU, toutes les heures de développeur futures rendues sur le projet sont classées sous *Main-d'œuvre Dev – Capital* (ex. : feuilles de temps avec ce type). De même, une facture de consultant pour la construction de la fonctionnalité est codée vers une catégorie de dépenses *Honoraires professionnels – Capital*. Résultat : à la fin du mois, le compte de résultat du projet ne montre aucune charge (les coûts s'accumulent dans le bilan). Par la suite, chaque mois, un journal d'amortissement automatisé réduit le solde de R&D différé et enregistre la charge d'amortissement dans le compte de résultat.

En configurant soigneusement les fonctionnalités de **Job Costing et d'amortissement** de NetSuite, un développeur SaaS peut mettre en œuvre l'ASU 2025-06 en douceur. Les outils intégrés de NetSuite – gestion de projet, budgets, suivi du temps et modèles d'amortissement – fournissent un cadre solide. Ce qui est essentiel, c'est de les exploiter pour différencier le travail capitalisé du travail passé en charges. Un livre blanc de NetSuite note que l'activation du Job Costing permet aux coûts de main-d'œuvre d'être « comptabilisés dans votre grand livre » par projet (Source: docs.oracle.com), ce qui est exactement la capacité nécessaire. En pratique, de nombreuses entreprises atteignent la conformité en créant des comptes de grand livre dédiés pour le « Développement de logiciel capitalisé » et en utilisant les types de dépenses de projet de NetSuite pour piloter les écritures comme décrit ci-dessus.

Analyse des données et preuves

Plusieurs points de données et enquêtes éclairent les effets de la comptabilisation des logiciels à usage interne et le contexte de l'ASU 2025-06 :

- Enquête sectorielle (Armanino, 2024)** : Une enquête exhaustive auprès des 100 plus grandes entreprises SaaS publiques américaines a révélé que **89 %** de ces entreprises capitalisent désormais les commissions de vente (ASC 340-40) – reflétant les effets de l'ASC 606 – contre seulement 22 % en 2017 (Source: www.armanino.com). En se concentrant sur le développement logiciel, **69 %** de ces entreprises SaaS capitalisent aujourd'hui les coûts logiciels internes (contre 62 % en 2017) (Source: www.armanino.com). Cela indique qu'une forte majorité d'entreprises SaaS utilisent déjà l'ASC 350-40 dans la pratique, de sorte que les clarifications de la nouvelle ASU codifieront une tendance existante plutôt que de renverser les pratiques. Fait intéressant, l'étude note également que les grandes et petites entreprises SaaS diffèrent peu en termes de taux de capitalisation : la capitalisation des coûts de développement est systématiquement élevée dans toutes les tranches de revenus.
- Analyse des dépôts auprès de la SEC** : Le rapport Armanino a analysé les informations fournies dans les formulaires 10-K pour déterminer les pratiques de capitalisation (Source: www.armanino.com). De nombreuses entreprises citent explicitement l'ASC 350-40 pour les coûts internes ou l'ASC 985-20 pour le développement externe (dans le cas d'Alphabet) (Source: www.wallstreetprep.com). Le 10-K de 2017 d'Alphabet, par exemple, indique qu'ils « passent en charges les coûts de développement logiciel » jusqu'à la faisabilité (capitalisant ainsi *presque rien* selon la norme 985-20), mais ils capitalisent les logiciels à usage interne une fois la phase préliminaire terminée (Source: www.wallstreetprep.com). Le 10-K d'AthenaHealth révèle l'inverse : elle capitalise une part substantielle du développement « athenaNet » selon la norme 350-40 (Source: www.wallstreetprep.com). Ces informations réelles soulignent à quel point le choix du modèle peut affecter radicalement les dépenses de R&D déclarées.
- Coût de la conformité (Commentaires de la SEC)** : Bien que les lettres de commentaires de la SEC sur la R&D logicielle soient apparemment rares (Source: www.armanino.com), les lettres sur la codification connexe (comme les commissions capitalisées) ont augmenté. L'enquête a reproduit des exemples de déclarations de la SEC qui sondent la manière dont les coûts tels que le développement commercial ou les commissions de renouvellement sont traités selon l'ASC 340-40 (Source: www.armanino.com) (Source: www.armanino.com). Bien que cela ne concerne pas directement l'ASC 350-40, cette tendance suggère que tout changement important dans la politique de capitalisation (par exemple, lors de l'adoption de l'ASU 2025-06) devrait être bien expliqué dans les dépôts pour éviter tout examen minutieux.
- Données fiscales** : Les données de conformité fiscale ne sont pas tabulées publiquement, mais les preuves anecdotiques sont frappantes. Les entreprises que nous avons examinées ont signalé des **impacts fiscaux substantiels la première année** en raison de la capitalisation des coûts de R&D selon l'article 174 du TCJA (Source: www.armanino.com). Pour de nombreuses entreprises SaaS, cela signifiait que des dizaines de millions de dollars de coûts ne pouvaient plus être immédiatement déduits, gonflant le revenu imposable en 2022. L'analyse d'Armanino avertit que seule une demi-année d'amortissement s'applique, donc une planification plus longue est nécessaire (Source: www.armanino.com). À l'avenir, l'ASU 2025-06 n'affecte pas le droit fiscal, mais l'alignement de la capitalisation PCGR avec les règles fiscales rend la question plus aiguë pour les décideurs.

- **Étude académique** : Une enquête de 2021 sur les pratiques comptables de l'industrie logicielle (Mulford & Roberts, 2021) a révélé une grande variance dans les taux de capitalisation et a noté que les différences de politique peuvent faire paraître deux entreprises similaires très différentes en termes de rentabilité (Source: www.wallstreetprep.com). Les auteurs ont conclu que sans une comptabilité cohérente, les comparaisons entre entreprises technologiques sont difficiles, faisant écho à la justification de cette ASU. Cela s'aligne avec le commentaire de Baker Tilly selon lequel l'ASU « réduit la diversité des pratiques » en améliorant l'opérabilité (Source: www.bakertilly.com).

Ces données et commentaires apportent un soutien solide aux changements du FASB. Ils montrent que (a) la plupart des entreprises SaaS *veulent* déjà capitaliser les coûts logiciels ; (b) les règles actuelles causent des incohérences ; (c) les changements fiscaux ont imposé une mentalité de capitalisation ; et (d) la nouvelle approche par seuil est considérée comme une évolution naturelle dans l'industrie (Source: www.armanino.com) (Source: www.wallstreetprep.com).

Études de cas et exemples

Pour illustrer la nouvelle guidance en action, nous considérons des scénarios représentatifs adaptés des propres exemples du FASB et de la pratique industrielle :

- **Mise en œuvre ERP (Développement sous contrat)** : *Service Co.* prévoit une mise à niveau ERP à l'échelle de l'entreprise. Elle consacre les mois de janvier à avril à la recherche et aux démonstrations des fournisseurs (phase préliminaire). Le **1er mai**, elle signe un contrat à prix fixe avec une société ERP pour mettre en œuvre et personnaliser un module. À ce stade, la direction a à la fois *autorisé le financement* (le contrat signé) et conclu qu'il est *probable d'achever* (la technologie est éprouvée et le périmètre fixé). Selon l'exemple 1 de l'ASU 2025-06 (*Service Co.*), les coûts capitalisables commencent le **1er mai** (Source: www.eidebailly.com). Tous les coûts postérieurs au 1er mai (personnel interne, frais de fournisseur) sont capitalisés ; par exemple, si un ingénieur passe 100 heures à coder après le 1er mai, ces salaires vont vers un compte d'actif. Les coûts antérieurs (recherche avant le 1er mai) restent passés en charges. Cela reflète la règle mise à jour : une fois le financement et la faisabilité établis, les coûts de développement s'accumulent au bilan (Source: www.eidebailly.com).
- **Nouvelle application mobile (Périmètre indéterminé)** : *Connect Co.* développe en interne une application mobile. Le **1er mars**, la direction approuve le budget et assigne une équipe, mais l'équipe est encore en train de définir « ce que l'application doit faire ». Initialement, ce projet présente une *incertitude significative* car les exigences fonctionnelles ne sont pas encore identifiées. Les critères de l'ASU 2025-06 ne sont pas encore remplis en mars : le financement est autorisé, mais « l'achèvement probable » est discutable en raison d'exigences non définies (Source: www.eidebailly.com). Par conséquent, les coûts du **1er mars** sont passés en charges. Au cours des mois suivants, des prototypes sont construits et les utilisateurs donnent leur avis. Au **1er novembre**, la direction finalise les fonctionnalités principales (résolvant l'incertitude). Comme le montre l'exemple 2 de la guidance, le 1er novembre, les seuils sont désormais satisfaits (les incertitudes liées à l'innovation ont disparu et le périmètre est fixé), de sorte que l'entreprise commence la capitalisation (Source: www.eidebailly.com). Tous les coûts éligibles à partir de novembre vont vers le compte d'actif ; les coûts avant cette date (et toute exploration de changements) engagés dans le développement restent passés en charges (Source: www.eidebailly.com). Ce scénario souligne que l'ASU permet aux entreprises de *commencer rétroactivement à capitaliser* une fois que les conditions sont réunies.
- **Développement de technologie innovante** : *Tech Co.* commence la conception d'un module d'IA révolutionnaire le **1er juillet 20X1**, en approuvant un budget et une équipe interne. Cependant, la fonctionnalité est réellement inédite. L'équipe rencontre des difficultés pendant des mois. Ce n'est que le **1er avril 20X3** que le codage prouve le concept et que les exigences se stabilisent (aucune modification majeure des fonctionnalités ni obstacle technique ne subsiste). Selon l'exemple 3 du FASB, l'exigence de « probabilité d'achèvement » est satisfaite le 1er avril 20X3 (Source: www.eidebailly.com). La direction s'était engagée le 1er juillet 20X1, mais jusqu'en avril 20X3, une incertitude significative persistait (caractéristiques inédites non résolues). Par conséquent, *la capitalisation débute le 1er avril 20X3*. Tous les coûts éligibles du 1er avril au 1er mai 20X3 (date à laquelle les tests sont terminés) sont capitalisés. Cela démontre l'accent mis par l'ASU 2025-06 sur la résolution des incertitudes : même avec un financement en place, aucun coût n'a été capitalisé tant que le projet n'a pas été dérisqué (Source: www.eidebailly.com).
- **Mise à niveau de fonctionnalité SaaS** : Prenons *SaaS Co.*, un fournisseur de logiciels par abonnement, qui développe une nouvelle fonctionnalité d'analyse majeure. L'équipe produit autorise le projet le **15 juin 20X2** mais poursuit le développement itératif (sprints Agile) jusqu'à la fin de l'année 20X2. En décembre, la fonctionnalité principale est entièrement définie et testée (aucun changement majeur à prévoir). Selon les anciennes règles, la capitalisation aurait pu être incertaine (car les phases se chevauchent), mais avec l'ASU 2025-06, le test s'applique en décembre : avec un financement approuvé et l'absence d'incertitude significative, *SaaS Co.* commence à capitaliser à partir de ce moment. NetSuite peut aider à suivre cela : l'équipe financière pourrait marquer le projet comme « Capitalisable au 01/12/20X2 » et reclasser les saisies

de temps facturées antérieurement (NetSuite permet la reclassification des temps historiques si nécessaire). Ce changement signifie que la moitié du coût du projet est capitalisée, l'amortissement débutant à partir de décembre. Si le projet était abandonné, tous les coûts seraient simplement passés en charges, et tout report précédent serait annulé.

Ces exemples illustrent comment l'ASU 2025-06 déplace la décision de capitalisation de « avons-nous atteint la phase de développement ? » à « nous sommes-nous engagés et le succès est-il probable ? ». Dans chaque cas, les *faits uniques* du projet ont déterminé la date exacte de capitalisation. Les entreprises réelles seront confrontées à des choix analogues : par exemple, le CTO d'une startup pourrait consigner dans les procès-verbaux des réunions le moment où le plan de développement s'est stabilisé ; une équipe comptable devrait archiver une note ou une liste de contrôle lorsqu'un projet répond pour la première fois aux critères d'autorisation et de résolution de l'incertitude. Les outils de gestion de projet de NetSuite peuvent soutenir ces flux de travail en enregistrant les dates clés du projet et l'achèvement des jalons, fournissant ainsi une piste d'audit.

Implications et orientations futures

Implications financières et stratégiques

Pour les développeurs SaaS et autres sociétés de logiciels, l'ASU 2025-06 a plusieurs implications clés :

- **Profil de résultat** : En permettant une capitalisation plus précoce dans les projets Agile, les entreprises déclareront des dépenses plus faibles (et un résultat avant impôts plus élevé) au début de la période de développement. La dépense différée est amortie ultérieurement, lissant l'impact. Les analystes doivent noter que certaines entreprises pourraient constater une augmentation ponctuelle des actifs incorporels immédiatement après l'adoption. La comparabilité entre les entreprises devrait toutefois s'améliorer, puisque la base de règles est désormais uniforme, quelle que soit la méthodologie.
- **Rapports sur les flux de trésorerie** : Les dépenses de R&D logicielle apparaîtront désormais plus clairement comme des flux de trésorerie liés aux **investissements**. Les entreprises doivent ajuster leurs analyses de flux de trésorerie en conséquence, car l'ASU exige explicitement que les sorties de fonds liées aux logiciels capitalisés soient classées comme des activités d'investissement (Source: www.armanino.com). Cela pourrait affecter des indicateurs tels que le flux de trésorerie disponible ou les ratios de dépenses d'investissement (capex) pour les entreprises technologiques. Les équipes financières doivent s'assurer que leur tableau des flux de trésorerie et leurs notes annexes sont alignés sur ce traitement.
- **Planification fiscale** : Compte tenu des changements de l'IRC §174, l'élargissement de la capitalisation des coûts de développement selon l'ASC 350-40 coïncide souvent avec la nécessité d'amortir ces coûts à des fins fiscales. Les entreprises doivent continuer à se coordonner avec leurs conseillers fiscaux sur le calendrier des déductions et l'utilisation des crédits. Étant donné que l'ASU 2025-06 permet des ajustements rétrospectifs, certaines entreprises pourraient choisir de l'appliquer de manière prospective (minimisant le risque de retraitement) et d'aligner progressivement les mesures fiscales et comptables. Les conseils d'Armanino soulignent que la gestion du revenu imposable et de la trésorerie (par exemple, l'ajustement des paiements d'impôts estimés) est désormais une question traitée au niveau du conseil d'administration dans de nombreuses entreprises SaaS (Source: www.armanino.com).
- **Politique et audits** : Les auditeurs examineront minutieusement la manière dont les entreprises interprètent les termes « probable » et « incertitude résolue ». Une documentation détaillée deviendra centrale pour les audits des dépenses de R&D. Les entités pourraient avoir besoin de mettre à jour leurs politiques comptables (dans les manuels internes) pour refléter le nouveau déclencheur de capitalisation. Le jugement accru signifie que les entreprises doivent être cohérentes et rigoureuses : une fois le seuil atteint pour un projet, tous les coûts similaires doivent être traités de la même manière. Les comités d'audit devraient être informés de la manière dont les budgets de projet, les évaluations des risques et les décisions de seuil sont pris, car Baker Tilly suggère que les conseils d'administration voudront « comprendre les changements » apportés par l'ASU 2025-06.
- **Investissement et valorisation** : Les investisseurs utilisent souvent des mesures telles que R&D/revenu ou R&D/valorisation pour évaluer les entreprises technologiques. L'ASU 2025-06 réduira les dépenses de R&D déclarées pendant un certain temps, gonflant potentiellement les indicateurs de rentabilité à court terme. Les analystes en valorisation devraient ajuster leurs modèles pour tenir compte des différences entre la R&D capitalisée et la R&D passée en charges. Sur la durée de vie d'un projet, cependant, le total des coûts reconnus sera le même ; il s'agit d'un décalage temporel. L'avantage clé est la fiabilité : les actifs incorporels capitalisés reflètent la valeur économique future (à condition que les entreprises maintiennent les projets).

Paysage comptable futur

L'ASU 2025-06 est une étape intermédiaire dans l'évolution plus large de la comptabilité des logiciels. Plusieurs développements connexes sont en cours :

- **Débat sur le modèle unique** : Il existe un débat de longue date (reflété dans l'invitation à commenter du FASB et les commentaires externes) sur la question de savoir si la comptabilité des logiciels internes et externes devrait converger. Les commentaires du directeur financier Andrew Hyde (Source: www.armanino.com) font écho à ce débat. Bien que le FASB n'ait pas unifié les modèles dans l'ASU 2025-06, il a fait passer à la fois les logiciels internes (350-40) et les sites Web (350-50) sous un cadre interne unique. Certaines voix de l'industrie poussent pour une convergence totale avec l'ASC 985-20 (l'« approche par modèle unique ») (Source: www.armanino.com). Si le FASB ou l'IASB poursuivaient un jour une norme plus unifiée, cela pourrait impliquer l'adoption de la notion de « faisabilité technologique » pour tous les logiciels. Un tel changement serait de grande envergure (et nécessiterait probablement un méga-projet distinct), mais les améliorations ciblées ici ne comblent pas cet écart.
- **Considérations internationales** : Selon les normes IFRS, les actifs incorporels générés en interne sont régis par l'IAS 38. L'IAS 38 exige que les entreprises ne capitalisent les coûts de développement qu'après avoir rempli certains critères (faisabilité technique, intention d'achèvement, etc.), et non au stade de la planification. En pratique, le modèle des IFRS est plus proche de l'ASC 985-20 que de l'ASC 350-40 ; de nombreuses juridictions n'autorisaient historiquement pas la capitalisation des logiciels développés en interne (traitant tout comme une dépense de R&D incorporelle jusqu'à l'achèvement). L'ASU 2025-06 *élargit* donc l'écart entre les PCGR (GAAP) et les IFRS, car les entreprises américaines continueront de capitaliser les développements internes selon l'ASC 350-40, tandis que les entreprises sous IFRS les amortiront. Toute démarche du FASB vers une approche unique pourrait nous rapprocher du traitement IFRS. Pour l'instant, les entreprises SaaS multinationales doivent concilier le fait qu'elles utilisent l'ASC 350-40 aux États-Unis et éventuellement un modèle plus strict à l'étranger.
- **Tendances du développement logiciel** : Les technologies émergentes comme l'IA pourraient modifier davantage la comptabilité. Armanino note que les coûts de formation interne à l'IA (pour créer de nouveaux logiciels) ne sont actuellement *pas* capitalisables – ils constituent des dépenses de R&D (Source: www.armanino.com). À mesure que l'IA sera davantage intégrée dans les produits, les directives pourraient évoluer (et les projets auront probablement des durées de vie utiles plus courtes). De plus, les modèles de livraison « continue » (logiciels permanents) remettent en question la définition de la portée du projet – les entreprises devront décider comment délimiter un « projet » d'un autre à des fins comptables. Les directives sur l'unité de compte, laissées au jugement dans l'ASU 2025-06, deviendront donc une question politique importante.
- **Futurs projets du FASB** : Le FASB a proposé une révision fin 2024 (avant l'ASU finale) sur les *coûts de développement initiaux* (Source: www.armanino.com). La substance de cette proposition reflète étroitement ce qui a été publié dans l'ASU 2025-06. À l'avenir, le FASB pourrait revoir des sujets connexes, tels que la présentation explicite des flux de trésorerie (déjà requise par la proposition (Source: www.armanino.com)) ou les définitions des actifs à long terme. Les parties prenantes pourraient également demander au FASB de considérer les logiciels comme une catégorie plus large d'actifs incorporels (par exemple, comment comptabiliser les logiciels développés en interne transférés ou vendus). Les cabinets d'audit partageront probablement les défis pratiques avec le FASB à mesure que les entreprises adopteront l'ASU 2025-06, guidant ainsi tout amendement futur.

Conclusion

La comptabilité des logiciels destinés à un usage interne est une question complexe depuis des décennies, mais l'ASU 2025-06 marque une modernisation significative. En **supprimant les étapes de validation obsolètes** et en introduisant un test de seuil clair, le FASB a aligné les PCGR américains sur les réalités du développement logiciel itératif. Pour les entreprises SaaS, cette mise à jour apporte une clarté indispensable : la capitalisation d'une nouvelle fonctionnalité ou plateforme dépend désormais de jugements délibérés sur le financement et l'incertitude, plutôt que sur l'achèvement parfois arbitraire d'une phase de planification. Cela devrait réduire la diversité des pratiques et améliorer la comparabilité entre les entreprises et les projets.

Néanmoins, la responsabilité incombe désormais à la direction d'exercer un jugement sain et une documentation robuste. Les équipes financières et informatiques doivent établir des processus pour déterminer et enregistrer exactement *quand* un projet répond pour la première fois aux critères « autorisés et probables ». Une intégration appropriée dans des systèmes comme NetSuite sera essentielle – en utilisant les fonctionnalités de comptabilité de projet et d'amortissement pour isoler et amortir les coûts capturés. Les exemples et les conseils ci-dessus illustrent à quoi cela pourrait ressembler en pratique.

Enfin, l'ASU 2025-06 ne reste pas isolée. Elle reflète des tendances plus larges : le débat sur la convergence dans la comptabilité des logiciels, les implications des changements de législation fiscale et la valorisation stratégique des actifs incorporels dans les entreprises technologiques. Comme l'a observé un ancien directeur financier SaaS, les entreprises devraient, en fin de compte, être en mesure de justifier la capitalisation d'un projet

selon *l'un ou l'autre* des modèles de logiciels à usage interne ou externe (Source: www.armanino.com). En attendant, l'ASU 2025-06 offre une voie plus rationalisée. Les développeurs SaaS et leurs équipes financières devraient profiter de cette occasion pour aligner leurs politiques et systèmes comptables sur la substance économique de leur développement de produits, en veillant à ce que NetSuite et d'autres outils capturent avec précision le nouveau traitement des coûts logiciels.

En conclusion, **le FASB ASU 2025-06 établit des règles plus exploitables et adaptables pour les logiciels à usage interne**. Il équilibre la flexibilité pour le développement moderne avec la responsabilité envers les actionnaires. En suivant ses directives — telles que détaillées dans ce rapport — les entreprises SaaS peuvent capitaliser en toute confiance les coûts éligibles dans NetSuite, rester conformes aux PCGR et présenter des états financiers plus clairs. Les nombreuses références citées ici (des communiqués du FASB aux livres blancs de l'industrie) démontrent la profondeur de l'analyse et le consensus autour de ce sujet. À l'avenir, les entreprises devraient surveiller les nouvelles annonces et les meilleures pratiques, mais peuvent être assurées que l'ASU 2025-06 représente l'orientation faisant autorité actuelle sur la capitalisation du développement logiciel dans NetSuite et au-delà.

Références : La littérature faisant autorité sur les PCGR et les analyses professionnelles ont été utilisées tout au long de ce document. Par exemple, le résumé et les illustrations de l'ASU 2025-06 du FASB (Source: rsmus.com) (Source: www.eidebailly.com), les livres blancs de RSM et Baker Tilly (Source: rsmus.com) (Source: www.bakertilly.com), l'analyse de Crowe LLP (Source: www.crowe.com), et la mise à jour DART de Deloitte (Source: dart.deloitte.com) fournissent le cadre technique. Les pratiques et perspectives de l'industrie sont tirées de sources telles que l'enquête SaaS d'Armanino (Source: www.armanino.com) (Source: www.armanino.com), les guides de Wall Street Prep et AccountingTools (Source: www.wallstreetprep.com) (Source: www.wallstreetprep.com), et la documentation NetSuite (Source: docs.oracle.com) (Source: netsuitedocumentation1.gitlab.io). Toutes les affirmations factuelles ci-dessus sont étayées par ces citations.

Étiquettes: fasb-asu-2025-06, asc-350-40, logiciel-usage-interne, capitalisation-logicielle, comptabilite-saas, comptabilite-netsuite, normes-gaap, couts-logiciels-immobilises

AVERTISSEMENT

Ce document est fourni à titre informatif uniquement. Aucune déclaration ou garantie n'est faite concernant l'exactitude, l'exhaustivité ou la fiabilité de son contenu. Toute utilisation de ces informations est à vos propres risques. Houseblend ne sera pas responsable des dommages découlant de l'utilisation de ce document. Ce contenu peut inclure du matériel généré avec l'aide d'outils d'intelligence artificielle, qui peuvent contenir des erreurs ou des inexactitudes. Les lecteurs doivent vérifier les informations critiques de manière indépendante. Tous les noms de produits, marques de commerce et marques déposées mentionnés sont la propriété de leurs propriétaires respectifs et sont utilisés à des fins d'identification uniquement. L'utilisation de ces noms n'implique pas l'approbation. Ce document ne constitue pas un conseil professionnel ou juridique. Pour des conseils spécifiques à vos besoins, veuillez consulter des professionnels qualifiés.