

Livre blanc

ERP
open source

Raphaël Valyi – Expert Technique – Smile

Version 1.0

Pour plus d'information :

www.smile.fr

Tél : 01 41 40 11 00

Mailto : sdcsmile.fr

ERP open source

PRÉAMBULE

Smile

Smile est une société d'ingénieurs experts dans la mise en œuvre de solutions open source et l'intégration de systèmes appuyés sur l'open source. Smile est membre de l'APRIL, l'association pour la promotion et la défense du logiciel libre.

Smile compte 250 collaborateurs (début 2008), ce qui en fait *la première société en France spécialisée dans l'open source*.

Depuis 2000, environ, Smile mène une action active de veille technologique qui lui permet de découvrir les produits les plus prometteurs de l'open source, de les qualifier et de les évaluer, de manière à proposer à ses clients les produits les plus aboutis, les plus robustes et les plus pérennes.

Cette démarche a donné lieu à toute une gamme de *livres blancs* couvrant différents domaines d'application. La gestion de contenus (2004), les portails (2005), la business intelligence (2006), les frameworks PHP (2007), la virtualisation (2007), et la gestion électronique de documents (2008), ainsi que les PGI/ERPs (2008).

Chacun de ces ouvrages présente une sélection des meilleures solutions open source dans le domaine considéré, leurs qualités respectives, ainsi que des retours d'expérience opérationnels.

Au fur et à mesure que des solutions open source solides gagnent de nouveaux domaines, Smile sera présent pour proposer à ses clients d'en bénéficier sans risque. Smile apparaît dans le paysage informatique français comme le prestataire intégrateur de choix pour accompagner les plus grandes entreprises dans l'adoption des meilleures solutions open source.

Ces dernières années, Smile a également étendu la gamme des services proposés. Depuis 2005, un département consulting accompagne nos clients, tant dans les phases d'avant-projet, en recherche de solutions, qu'en accompagnement de projet. Depuis 2000, Smile dispose d'un studio graphique, devenu en 2007 Agence Media Interactive, proposant outre la création graphique, une expertise e-marketing, éditoriale, et interfaces riches. Smile dispose aussi d'une agence spécialisée dans la Tierce Maintenance Applicative, le support et l'exploitation des applications.

Enfin, Smile est implanté à Paris, Lyon, Nantes, Bordeaux et Montpellier.

ERP open source

Quelques références**Intranets - Extranets**

Société Générale, Caisse d'Épargne, Bureau Veritas, Commissariat à l'Energie Atomique, Visual, Vega Finance, Camif, Lynxial, RATP, SPIE, Sonacotra, Faceo, CNRS, AmecSpie, Château de Versailles, Banque PSA Finance, Groupe Moniteur, CIDJ, CIRAD, Bureau Veritas, Ministère de l'Environnement, JCDecaux, Ministère du Tourisme, DIREN PACA, SAS, Institut National de l'Audiovisuel, Cogedim, Ecureuil Gestion, IRP-Auto, AFNOR, Conseil Régional Ile de France, Verspieren, Zodiac, OSEO, Prolea, Conseil Général de la Côte d'Or, IPSOS, Bouygues Telecom, Pimki Diramode, Prisma Presse, SANEF, INRA, HEC, ArjoWiggins

Internet, Portails et e-Commerce

cadreemploi.fr, chocolat.nestle.fr, creditlyonnais.fr, explorimmo.com, meilleurtaux.com, cogedim.fr, capem.fr, editions-cigale.com, hotels-exclusive.com, souriau.com, pci.fr, dsv-cea.fr, egide.asso.fr, osmoz.com, spie.fr, nec.fr, sogeposte.fr, nouvelles-frontieres.fr, metro.fr, stein-heurtey-services.fr, bipm.org, buitoni.fr, aviation-register.com, cci.fr, schneider-electric.com, calypso.tm.fr, inra.fr, cnil.fr, longchamp.com, aesn.fr, Dassault Systemes 3ds.com, croix-rouge.fr, worldwatercouncil.org, projetif.fr, editionsbussiere.com, glamour.com, fratel.org, tiru.fr, faurecia.com, cidil.fr, prolea.fr, ETS Europe, ecofi.fr, credit-cooperatif.fr, odit-france.fr, pompiersdefrance.org, watermonitoringaliance.net, bloom.com, meddispar.com, nmmedical.fr, medistore.fr, Yves Rocher, jcdecaux.com, cg21.fr, Bureau Veritas veristar.com, voyages-sncf.fr, eurostar.com, AON, OSEO, cea.fr, eaufrance.fr, banquepsafinance.com, nationalgeographic.fr, idtgv.fr, prismapub.com, Bouygues Construction, Hachette Filipacchi Media, ELLE.fr, femmeactuelle.fr, AnnoncesJaunes.fr

Applications métier, systèmes documentaires, business intelligence

Renault, Le Figaro, Sucden, Capri, Libération, Société Générale, Ministère de l'Emploi, CNOUS, Neopost Industries, ARC, Laboratoires Merck, Egide, Bureau Veritas, ATEL-Hotels, Exclusive Hotels, Ministère du Tourisme, Groupe Moniteur, Verspieren, Caisse d'Épargne, AFNOR, Souriau, MTV, Capem, Institut Mutualiste Montsouris, Dassault Systemes, Gaz de France, CFRT, Zodiac, Croix-Rouge Française, Centre d'Information de la Jeunesse (CIDJ), Pierre Audoin Consultants, EDF, Conseil Régional de Picardie, Leroy Merlin, Pierre Audoin Consultants, Renault F1, l'INRIA, Primagaz, Véolia Propreté, Union de la Coopération Forestière Française, Ministère Belge de la Communauté Française

Ce livre blanc

Dans le contexte actuel d'une compétition internationale féroce, les entreprises doivent constamment rationaliser leur pilotage. Parce qu'il centralise et présente les données de façon pertinente et parce qu'il délègue exactement les bons process aux bons acteurs, l'ERP est indéniablement l'outil de ce pilotage.

Pourtant, traditionnellement les ERP ont toujours été réservés aux grandes entreprises et à une poignée de très grands éditeurs. Dès lors, les PME et TPE n'y avaient pas accès ou alors devaient se contenter de plus modestes logiciels de comptabilité et gestion commerciale, pas d'un vrai ERP apte à s'adapter finement aux structures de données et process de gestion ou de production spécifiques à chaque entreprise.

Aujourd'hui, avec l'essor indéniable des logiciels open source dans tous les domaines, **l'offre d'ERP open source arrive enfin à maturité. Que vaut-elle et quels en sont les meilleurs outils?**

Smile a abordé cette question en **totale indépendance de tout éditeur, depuis septembre 2007**. Nous avons eu comme ambition d'avoir une couverture exhaustive: c'est-à-dire de passer au crible (fonctionnel, technique, communautaire) pratiquement tous les produits qui se disent ERP open source, quitte à faire un premier tri impitoyable, afin d'être sûr de **ne rater aucune bonne solution adaptée au marché français, mais aussi dans le souci d'une pérennité de la solution**.

La portée de cette étude nous semble d'autant plus large qu'il est très difficile de créer un nouvel ERP en moins de 3 ans tant il s'agit d'un produit complexe. Ceci veut dire que s'il y a bien des dynamiques de fond, les produits, même ceux des prochaines années, sont déjà clairement identifiés et portent souvent dans leur gènes technologiques les potentialités ou limites de leur trajectoire future tant il est vrai que pour l'ERP, **le fonctionnel est si complexe qu'il exige forcément d'abord un bon terreau conceptuel pour se développer de façon durable**.

Ce n'est qu'à l'issue de ce passage au crible multi-critères que nous avons retenu une shortlist composée, on peut déjà le dire de: TinyERP (rebaptisé OpenERP), Openbravo, Compierre, Adempierre, Neogia et ERP5. C'est ainsi que nous illustrerons souvent nos explications fonctionnelles par les réponses concrètes apportées ou non par ces ERP.

Cependant tous ces ERP ne se valent pas, loin s'en faut. Nous vous expliquerons pourquoi au final Smile a décidé de retenir prioritairement **TinyERP (OpenERP), Openbravo et Neogia**, comme des solutions compétitives, fiables et pérennes. Car ne l'oublions pas, un ERP installé aujourd'hui devra encore répondre présent et s'ouvrir à de nouvelles fonctionnalités à la hauteur de vos exigences dans 5 ans...

ERP open source

A l'issue de cette étude, Smile a aussi décidé de signer des partenariats avec TinyERP et Openbravo pour appuyer sa force de frappe. Pourtant, il est très important de noter que notre analyse est antérieure à ces partenariats qui n'ont été signés qu'en Février 2008.

Nous vous expliquerons aussi pourquoi Smile n'a pas décidé de s'engager ni avec Compiere ni ERP5, solutions fonctionnelles et techniques honorables mais qui n'offrent pas en l'état d'ouverture suffisante pour l'intégration par de nouvelles tierces parties telles que Smile.

Smile n'a malgré tout pas la connaissance exhaustive des fonctionnalités des 6 produits de la shortlist étudiée tant ils sont complexes et les éditeurs pas toujours enclins à produire de la documentation ouverte (cas de Compiere par exemple). **Mais, lorsque nous le savions, nous nous sommes toujours efforcés d'explicitier la solution ou au contraire l'absence de solution pour chaque ERP. Au final, l'important pour nous était aussi d'évaluer et de présenter les solutions finalement préconisées.**

Table des matières

PRÉAMBULE.....	2
SMILE.....	2
QUELQUES RÉFÉRENCES	3
CE LIVRE BLANC.....	4
INTRODUCTION.....	8
A QUOI SERT UN ERP ?.....	8
VALEUR AJOUTÉE D'UN ERP OPEN SOURCE EN PARTICULIER.....	12
L'ERP LIBRE, UN PRODUIT COMPLEXE NÉ DE CONJONCTURES OPEN SOURCE PARTICULIÈRES.....	16
QUAND PEUT-ON ENVISAGER UN ERP OPEN SOURCE?	24
QUELS PRODUITS COMMERCIAUX SONT CONCURRENCÉS PAR LES ERP OPEN SOURCE PRÉSENTÉS?	27
LES DOMAINES FONCTIONNELS FONDAMENTAUX DES ERP	29
PARAMÉTRAGE DE BASE, PRODUITS ET PARTENAIRES.....	29
PRICING.....	31
VENTES ET GESTION DES PARTENAIRES (CRM)	33
VENTE PAR PLATEFORME E-COMMERCE.....	37
VENTE PAR POINTS DE VENTE (VENTE EN CAISSE).....	38
GESTION DES ACHATS	40
GESTION DES STOCKS	41
GESTION DE LA PRODUCTION	43
GESTION DE PROJETS	48
COMPTABILITÉ	50
BUSINESS INTELLIGENCE	54
CONCEPTS TRANSVERSAUX D'UNE PLATEFORME ERP.....	56
STRUCTURE ÉLÉMENTAIRE DE L'INTERFACE D'UN ERP.....	56
LE SYSTÈME DE RÔLES	58
MULTI-SOCIÉTÉS	58
CLIENT WEB OU CLIENT LOURD?.....	59
REPORTING.....	61
ERGONOMIE	61
TECHNOLOGIE ET ARCHITECTURES	63
INTERFACES MÉTA-GÉNÉRÉES À PARTIR DES DONNÉES.....	64
BASE DE DONNÉES RELATIONNELLE CONNUE ET ROBUSTE.....	64
TENUE EN CHARGE ET SCALABILITÉ.....	66
DICIONNAIRE DE STRUCTURE DE DONNÉES DYNAMIQUE.....	67
CODE MÉTIER BASÉ SUR LE PARADIGME OBJET.....	68
WEBSERVICES (SOA) FOURNIS EN STANDARD PAR LA PLATEFORME.....	69
MODÉLISATION BPM DE HAUT NIVEAU.....	70
MODULARITÉ DE LA PLATEFORME.....	72
PREMIERES CRITÈRES DE SÉLECTION.....	75
TAILLE ET DYNAMIQUE CRITIQUE.....	75
ARCHITECTURE TECHNIQUE MINIMALE.....	76
SÉLECTION DES ERP OPEN SOURCE SATISFAISANT À NOS CRITÈRES	77
ÉTABLISSEMENT D'UNE 'SHORTLIST'.....	81
SÉLECTION RÉDUITE DES MEILLEURS ERP	82
PROFIL MACROSCOPIQUE DES ERP	
PRÉ-SÉLECTIONNÉS.....	82
PROFIL PAR CARACTÉRISTIQUES GÉNÉRALES.....	82
PROFIL PAR DOMAINE FONCTIONNEL.....	87
PROFIL PAR SECTEUR ET TAILLE D'ENTREPRISE.....	88

ERP open source

CHOIX ENTRE COMPIERE, ADEMPIERE ET OPENBRAVO.....	93
LES LIMITES D'ERP5.....	96
RÉCAPITULATIF SUR OPENBRAVO.....	98
RÉCAPITULATIF SUR TINYERP (OPENERP)	104
RÉCAPITULATIF SUR NEOGIA.....	109
SYNTHÈSE.....	115

INTRODUCTION

A quoi sert un ERP ?

Un ERP (Enterprise Resource Planning) ou encore en français PGI (Progiciel de Gestion Intégré) est un **référentiel central** de gestion de toutes les données d'une entreprise, autour duquel gravitent un ensemble de modules fonctionnels qui effectuent les traitements métier sur ces données, par exemple les traitements liés à la vente d'un produit.

Plus précisément, ce référentiel a pour vocation première d'organiser et de consigner chaque fait lié à un flux financier de l'entreprise afin d'automatiser la gestion comptable d'une part et d'offrir des métriques d'aide au pilotage des ventes et de la production, à la décision stratégique, d'autre part. Il permet ainsi une meilleure synergie des acteurs, du commercial au technicien de production en passant par le directeur financier ou le logisticien.

Les grands modules généralistes de l'ERP

Si les limites du périmètre fonctionnel peuvent varier, on considère en général qu'un ERP prend en charge les domaines suivants.

Comptabilité

Il s'agit au moins de la comptabilité analytique qui s'appuie éventuellement sur une infrastructure de business intelligence embarquée par l'ERP. Certains ERP gèrent aussi la comptabilité générale française, mais à l'heure actuelle leur mise en oeuvre correcte nécessite encore des paramétrages assez intenses. Néanmoins, un **pont comptable d'export d'écritures** peut être mis en place pour utiliser une gestion comptable abordable mais éprouvée tout en conservant les outils d'analyse, de facturation et de gestion commerciale de l'ERP open source.

Achats:

Le module d'achat permet de gérer les transactions d'achat et écritures comptables associées, mais aussi les approvisionnements selon des politiques à paramétrer et/ou selon le calcul des besoins déterminés par la gestion de production.

ERP open source

Ventes

Ecritures comptables des ventes, mais aussi: règles de pricing, devis, factures, paiements... Certains ERP, vont aussi très loin dans le **CRM** (Customer Relation Management) ou GRC (Gestion de la Relation Client)

Dans certains cas, l'ERP peut intégrer une plateforme d'e-commerce native. Mais plus généralement l'ERP disposera de webservices et/ou connecteurs SQL permettant d'interfacer des logiciels d'e-commerce standard.

Parfois encore, les ERP s'interfaçent nativement avec des solutions de ventes en caisse POS (Point Of Sale) ou encore Point de Vente en français.

Stocks et inventaires

Il s'agit essentiellement de gérer des politiques d'approvisionnement de stocks en fonction des ventes. On parle ici de **SCM** (Supply Chain Management), ou en français GCL, (Gestion de la Chaîne Logistique). Les bons ERP sont multi-entrepôts et comptabilisent tous les mouvements de stock, c'est-à-dire font de la gestion de stock à *double entrées*. La gestion des stocks permet aussi généralement la traçabilité.

Gestion de production: MRP

MRP signifie Manufacturing Resource Planning, soit en français GPP: Gestion et Planification de la Production.

On distingue:

- La gestion de production simple, la **MRP1**, qui ne fait que calculer les besoins en fonction des commandes et laisse ensuite soin à l'administrateur de vérifier qu'on n'atteint pas de goulot d'étranglement, quitte à re-planifier des opérations le cas échéant;
- La gestion de production à capacité finie, ou **MRP2**, qui est capable d'optimiser la production au maximum de sa charge selon un ensemble de contraintes paramétrables.

Les modules moins traditionnels

De façon moins systématique, on trouve encore dans certains ERP les fonctionnalités suivantes.

Gestion de projet

Répartitions de tâches à des personnes, diagrammes de GANTT... C'est en fait la gestion de production des sociétés de service. Bien que moins souvent implémentée pour des raisons historiques, la gestion de projet est néanmoins essentielle pour les entreprises de service.

ERP open source

Ressources humaines

Le périmètre du module *ressources humaines* peut varier de la gestion des emplois du temps, au recrutement, en passant par la gestion de la paie. A noter que les modules de paie sont très rares à cause du morcellement législatif d'une part et de la mise en jeu de données très confidentielles d'autre part.

CMMS (Computerized Maintenance Management System)

En français, c'est la GMAO: Gestion de la Maintenance Assistée par ordinateur. Ce type de module sert de référentiel des opérations de maintenance et n'est pas très complexe. On pourra assez facilement l'ajouter s'il n'est pas offert nativement.

D'une manière générale, les ERP open source sont assez solides sur ces grands modules généralistes, même si la gestion de production complexe est plus ou moins bien traitée selon les produits. Il est facile de comprendre que l'open source bénéficie ici à plein d'un effet de mutualisation des développements et des tests sur les modules les plus largement utilisés.

La verticalisation

Viennent ensuite des modules spécifiques adaptés à des secteurs d'activités bien particulier (fabrication, distribution, textile, agro-alimentaire, chimie...). On appelle ceci la **verticalisation** d'un ERP. Chaque ERP possède ainsi une liste plus ou moins large de ces modules verticaux. Cette verticalisation s'accompagne généralement d'un paramétrage des interfaces utilisateurs qui adoptent le *jargon* du métier concerné et rend l'utilisation plus intuitive ou plus conforme aux interfaces utilisées historiquement.

Lorsqu'un ERP libre n'a pas déjà nativement la verticalisation sectorielle recherchée, on peut en général la ré-implémenter avec un coût qui dépend essentiellement de la souplesse autorisée par la technologie de l'ERP et du dynamisme de son activité communautaire, c'est-à-dire du nombre de modules libres similaires et du nombre de personnes susceptibles d'aider sur les forums.

Il est important néanmoins de relativiser l'importance de cette verticalisation qui est encore essentiellement l'apanage des solutions commerciales:

- Elle répond parfois plus à des critères commerciaux de sectorisation du marché qu'à des critères fonctionnels.
- Dans le contexte de la mondialisation et d'une concurrence de plus en plus forte, la flexibilité, la capacité à s'adapter à des

ERP open source

modèles changeants, est souvent plus importante que la conformité aux usages anciens du métier.

Les plus avancés des ERP présentés ici arrivent à un stade de maturité tel qu'ils abordent cette phase de verticalisation. Pour certains, la modularité de la plateforme et les plugins verticaux sont déjà largement en place, il ne leur reste alors essentiellement plus qu'une phase de packaging marketing pour apporter de la lisibilité à ces solutions verticalisées. Sur d'autres, moins extensibles, la spécialisation des modules représentera un développement plus conséquent.

Un ERP ne doit pas conquérir un secteur au détriment des autres. La verticalisation passe donc par une plateforme technique de qualité qui offre en premier lieu une modularité suffisante.

Modularité applicative, mais référentiel transactionnel unique

Techniquement parlant, alors que nous venons de vanter les mérites de la modularité applicative, l'usage d'un moteur de base de données relationnel central et monolithique est justifié par le besoin d'intégrité transactionnelle des données d'entreprises telles que les inventaires, mais plus généralement de tous les flux financiers.

En revanche, les applications métiers qui gravitent autour de ce référentiel doivent être modulaires. A ce titre, les ERP que nous allons mettre en avant sont compatibles, de façon plus ou moins native selon le produit, avec une approche SOA (Service Oriented Architecture).

Retour sur investissement

Le retour sur investissement d'un ERP en général est souvent admis par principe mais bien difficile à chiffrer car il faudrait mener des études comparatives sur des cas réels et de longues périodes. Or d'une part on manque de cas similaires ne différant que par l'ERP et d'autre part, il est difficile d'isoler les conséquences exactes d'un ERP des facteurs d'ordre macro-économiques et concurrentiels.

Les clés du retour sur investissement d'un ERP sont néanmoins:

- L'économie de ressources opérationnelles, grâce à l'élimination des efforts redondants et des données dupliquées.
- L'augmentation de la capacité de ventes grâce à la rationalisation de la production.
- L'amélioration de la réactivité de la production vis à vis de la demande ou au contraire la meilleure adéquation de la stratégie marketing en fonction des capacités de production.
- L'augmentation du volume des ventes par l'amélioration de la relation clients et une politique de prix optimisée dynamiquement voire en temps réel.

ERP open source

- L'optimisation des approvisionnements: diminution des stocks inutiles et meilleures stratégies d'achats.
- Une plus grande satisfaction du personnel, des clients et des fournisseurs, grâce à des processus plus adaptés et moins de tâches rébarbatives.

Enfin, certains clients ou investisseurs exigent la présence d'un ERP pour travailler avec une entreprise.

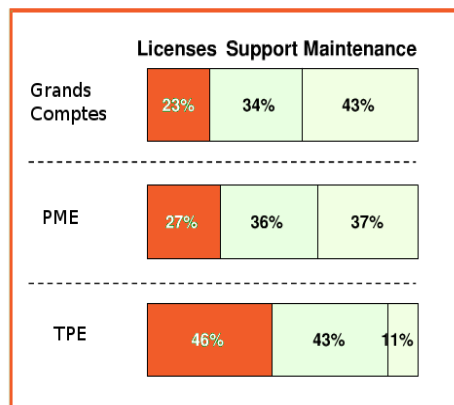
Valeur ajoutée d'un ERP open source en particulier

D'une manière générale, comme avec toute famille de produits open source on peut s'attendre à des économies de licence en installant un ERP open source. En première approche, l'ERP étant un progiciel complexe, les coûts d'intégration et de maintenance rendent cette économie directe de licence modérée au regard du coût total de possession de l'ERP. Ainsi, **l'économie d'une licence propriétaire représenterait entre 25% et 50% du coût total de possession** (incluant intégration, support et maintenance) **à périmètre équivalent**.

Toutefois, cette économie est d'autant plus difficile à estimer que les licences des ERP haut de gamme sont vendues à des prix confidentiels très variables et qui s'inscrivent dans une politique marketing à long terme. Quant aux intégrations, les chiffrages des propositions commerciales concernent en général un très petits périmètres alors que sur l'essentiel du périmètre aucun intégrateur n'acceptera de s'engager sur des prestations au forfait.

Mais comme nous allons le voir, au delà même des coûts, un **ERP open source permet de viser une meilleure couverture du besoin** de par sa plus grande extensibilité.

ERP open source



Source: IDC, 2004.

ILLUSTRATION 1 : DISTRIBUTION DES COÛTS DE LICENCE, DE SUPPORT ET DE MAINTENANCE SUR LES ERP COMMERCIAUX

Précisons néanmoins que si on doit re-développer en partie des modules spécifiques non présents nativement sur un ERP libre, alors une partie de l'économie du prix de licence sera absorbée dans ces développements. L'appréciation de cet effort à fournir et le choix de l'ERP libre le plus approprié sont donc cruciaux et ce livre blanc est là pour vous aider à mesurer cet effort.

Cependant, le bénéfice peut aller bien au delà de l'économie directe de licence. En effet, une forte proportion des intégrations d'ERP propriétaires donne lieu à une forte insatisfaction des clients.

Ainsi, selon selon une étude menée par Standish Group en 2000, des projets ERP commerciaux:

- Seuls 28% respectent le budget, dans les temps et satisfont aux besoins
- 49% sont des échecs soit en terme de budget, de délai ou de fonctionnalités
- 23% sont abandonnés avant la finalisation

D'après une autre étude encore de MC Kinsey, seul 5% des installations d'ERP sont de véritables succès!

Outre le point essentiel qu'est le management du changement lié à tout projet ERP mais qui dépasse le périmètre de ce livre blanc, voyons les points où l'ERP libre se distingue et permet d'évaluer et d'éviter les risques de dérapage budgétaire par sa plus grande transparence:

- **Souplesse par l'ouverture:** même si les éditeurs d'ERP propriétaires donnent un certain accès à leur ERP via des webservices, c'est toujours un accès très limité dans lequel l'ERP reste une boîte noire. Parfois c'est tout bonnement insuffisant pour permettre d'intégrer des

ERP open source

extensions. Sans compter que cet isolement rend en général la recherche de documentation de piètre efficacité par rapport à la recherche sur le web de ressources open source communautaires. Dans tous les cas, avec l'ERP libre, c'est l'ensemble du code qui est ouvert et pas seulement ses webservice. En conséquence, votre personnel informatique devrait pouvoir se mettre à niveau pour résoudre votre problème. Au pire vous trouveriez facilement un intégrateur, d'autant plus facilement que celui-ci aura pu lui-même s'auto-former aisément. Au contraire, sur des technologies fermées comme ABAP de SAP, trouver un expert est particulièrement difficile.

- **Souplesse par respect des standards:** Les ERP propriétaires ont tendance à respecter moins de standards. Certaines briques open sources ne peuvent pas non plus être re-distribuées avec des offres propriétaires, c'est notamment le cas des logiciels sous licence GPL dont vous devrez alors financer des équivalents propriétaires sans réel apport de valeur ajoutée par rapport à l'open source. Le fossé se creusant inéluctablement entre le monde du logiciel libre et les éditeurs propriétaires, comme comme ces derniers ne peuvent assumer tous les coûts d'implémentation de standards, ils en implémentent peu ou utilisent leurs propres protocoles qui n'ont pas de validité passée la frontière de l'éditeur en question; on parle ici alors de « vendor lock-in » (verrouillage par le fournisseur).
- **Qualité:** souvent, l'open source se distingue par une qualité technique supérieure. En effet, le degré de qualité d'un logiciel libre est mesurable par son code ouvert et par les réactions de la communauté d'utilisateurs. Et ainsi seuls les projets open source de qualité se développent. De plus, contrairement aux éditeurs de logiciel propriétaire, les éditeurs open source sont plutôt engagés dans ces défis techniques mesurables que sur l'exploitation au maximum de leurs contrats existants.
- **C'est votre ERP qui s'adapte à votre entreprise et non l'inverse:** du fait de leurs limitations en terme de flexibilité et de business model, les éditeurs d'ERP propriétaires l'érigent en règle: si l'ERP ne convient pas à vos process, changez vos process! Ils justifient cette doctrine en affirmant connaître votre métier mieux que vous. C'est parfois vrai, mais souvent les entreprises ont aussi des avantages compétitifs dans l'implémentation de certains process et dans ces cas, un ERP libre sera un instrument de pilotage plus fin.
- **Infrastructure éditeur en adéquation avec les budgets PME/TPE:** les gros éditeurs d'ERP se sont développés pour des grands comptes qui disposaient d'importants budgets informatiques. En conséquence, ces éditeurs possèdent un modèle d'infrastructure très coûteux qui leur interdit de s'adapter à la réalité économique des TPE et PME. Pour être rentables sur ces marchés, ils devraient vendre de plus gros volumes et totalement remettre en cause leur infrastructure. En corollaire, la nature technique même des ERP dédiés aux grands comptes obligera aussi à des coûts d'intégration qui ne sauront convenir aux PME ou TPE. C'est bien pourquoi même les éditeurs

ERP open source

s'efforcent finalement de chercher de nouvelles solutions techniques aptes à faire chuter les coûts d'intégration pour séduire ce segment de marché. Mais ces nouveaux produits, tels que Business One de SAP sont en général trop limités.

- **Pas de coûts cachés:** Malgré tout, de nombreux éditeurs d'ERP propriétaires offrent des prix cassés pour des packages limités, notamment en nombre d'utilisateurs. Ces politiques tarifaires ont souvent comme seul objectif de s'approprier de nouveaux clients pour les rendre ensuite captifs par de coûteuses politiques commerciales. Ensuite, lorsque l'entreprise grossit et si on veut plus d'utilisateurs sur l'ERP, les coûts peuvent très rapidement s'envoler à la juste mesure du degré de captivité du client. Seule une licence open source approuvée OSI et un code de bonne qualité (dont vous ne serez pas techniquement prisonnier) vous protègent réellement contre ces situations.
- **Pérennité des développements spécifiques:** les éditeurs d'ERP propriétaires s'efforcent de garder la maîtrise sur l'ensemble des modules fonctionnels dont vous avez besoin. Sans standard et sous le coup de la licence, difficile de vous garantir que vos propres développements resteront compatibles ou que vous allez pouvoir mettre en concurrence différents prestataires pour développer certains modules.
- **Support pragmatique:** le support est-il vraiment au rendez-vous avec les éditeurs propriétaires? Rappelez-vous des coûts d'infrastructure: en réalité un support approprié coûte très cher à fournir et les éditeurs, forts d'une relation totalement asymétrique auront tendance à rogner dessus. Par pragmatisme, mieux vaut alors bénéficier de la documentation communautaire d'ERP libres ou encore du support concurrentiel des prestataires associés à cette offre.
- **Vous n'êtes pas une « vache à lait »:** Les éditeurs d'ERP propriétaires ne vous apportent aucune garantie que les développements spécifiques que vous allez payer seront réellement des développements spécifiques: ils peuvent être tentés de vous faire payer des développements qui ont déjà été amortis par d'autres clients. Au contraire une licence open source telle que la GPL vous protège contre ces abus: un développement spécifique doit être reversé gratuitement à la communauté pour être distribué.
- **Indépendance vis-à-vis du destin d'un éditeur:** Qu'arrive-t'il si votre éditeur d'ERP propriétaire fait faillite? Généralement, il y a bien des clauses qui imposent de libérer le code source. Mais ceci n'est pas d'une très grande utilité: la solution survivra très peu de temps en l'absence de support, documentation appropriée, et gouvernance communautaire. Au contraire, un ERP libre avec un minimum de qualité technique est un choix pérenne car même si un prestataire faisait faillite, d'autres seraient déjà prêts à prendre le relais. Si un

ERP open source

éditeur libre ne sait pas rassurer, il est probable qu'un fork¹ se développe avec une base plus saine. Ceci fera d'ailleurs pression sur l'éditeur historique pour qu'il corrige éventuellement sa politique, le client a donc un moyen de pression pro-actif sur l'éditeur et dans l'idéal tout sera résolu sans heurt. (Lire plus loin le fork Adempiere de Compiere)

- **Indépendance vis-à-vis des fusions/acquisitions:** Un éditeur d'ERP propriétaire peut bien être racheté à l'instar de J.D. Edwards par Peoplesoft puis PeopleSoft par Oracle. Dans ce cas, il est très probable que le support de l'ERP d'origine se tarisse avec diverses pressions de la part de l'éditeur pour que vous soyez forcé de migrer dans le cas où ce sont des concurrents que l'entreprise a racheté ou encore d'acheter des modules initialement non prévus dans le cas où les entreprises sont complémentaires comme dans le cas d'Oracle applications.
- **Le discours commercial est moins biaisé par des intérêts propres:** Devant la multiplicité des acteurs impliqués dans la vente d'ERP propriétaires, êtes-vous sûrs d'être toujours libre de vos choix? Au niveau des contrats commerciaux entre éditeurs et revendeurs, c'est rarement la qualité technique et l'adéquation fonctionnelle qui préside, mais plutôt des petits accords sur les objectifs, contrats d'exclusivité sectorielle ou géographique etc. Pas de cela avec un ERP libre.
- **Vous upgradez librement:** quand dois-je upgrader? Avec un ERP libre vous n'upgradez que lorsque vous le souhaitez. Et si vous le faites, vous pouvez toujours le faire par vos soins gratuitement ou alors au meilleur prix en mettant des intégrateurs en concurrence, en tout cas sans coût de licence. Au contraire, avec un éditeur propriétaire, ces coûts de mise à jour peuvent s'envoler à la juste mesure de votre degré de captivité.

**L'ERP libre, un produit complexe né
de conjonctures open source
particulières**

Dans le chapitre précédent, on a affirmé qu'un ERP libre a un coût total de possession sensiblement inférieur à celui d'un ERP propriétaire. Pourtant, par sa complexité technique et fonctionnelle, un ERP à un coût de développement que quelqu'un doit bien financer. **Au contraire d'autres logiciels plus simples comme des outils de publication de contenu web, l'ERP libre ne résulte que très marginalement de travail bénévole** fait par quelques hackers le soir, du moins y a-t-il un cœur fonctionnel cohérent qui nécessite l'investissement et tout le poids d'un

¹Un fork est la décision d'une communauté de poursuivre elle-même le développement d'un produit alternatif issu à l'origine de la même base de code mais en s'affranchissant désormais de la tutelle de l'éditeur historique, ainsi que le garantit la licence open source.

ERP open source

éditeur. Il y a aussi des modules fonctionnels qui doivent suivre les évolutions législatives et dont la maintenance nécessite encore plutôt de la pérennité financière que de la passion bénévole. Il faut savoir que les rivaux commerciaux tels que SAP investissent eux plusieurs dizaines de millions d'euros sur leurs ERP!

Comment expliquer alors les économies annoncées ?

- **Mise en compétition favorisant les économies:** en premier lieu, l'open source opère une chasse au gaspillage impitoyable: tout coût n'apportant pas de valeur ajoutée à l'utilisateur final est gommé par un mécanisme de sélection darwinienne: qu'un acteur exagère le poids de son prélèvement et essaie de tirer une quelconque rente d'une situation de monopole et la licence open source lui garantit le **fork** assuré qui saura s'engouffrer dans cette nouvelle niche pour offrir un service plus compétitif².
- **Le marketing viral remplace les coûts marketing traditionnels:** en second lieu, un logiciel open source peut largement s'affranchir des moyens marketing généralement engloutis pour faire la promotion de produits commerciaux. Dans un bon logiciel open source, le bouche à oreille joue à plein: toute qualité (ou défaut) est immédiatement et viralemment diffusée dans sa communauté d'utilisateurs qui deviennent alors les prescripteurs de ce marketing 'low cost'. Et les experts peuvent à tout instant vérifier l'adéquation du code source avec les bénéfices annoncés, interdisant toute campagne marketing qui ne correspondrait pas à une réalité.
- **Des efforts mutualisés:** enfin, avec le modèle open source, l'ouverture du code et le respect de protocoles standards donne lieu à un véritable écosystème d'acteurs qui mutualisent leurs contributions. Ainsi, plusieurs acteurs peuvent se fédérer pour financer le développement d'un module de paie ou la verticalisation d'un secteur d'activité bien spécifique (voir le mécanisme de « shared funding » mis en place chez TinyERP et récemment adopté également chez Openbravo).

Mais malgré toutes ces économies, **un ERP libre doit bel et bien disposer d'un investissement de R&D initial pour incuber.** Les modalités de cette incubation ne sont pas anodines et influent beaucoup sur les futurs coûts d'évolution fonctionnelle d'un ERP. Ainsi, nous pouvons caractériser les ERP en 2 grandes familles selon leur type d'incubation.

a) Les ERP libres coûteux, aujourd'hui matures

En premier lieu, nous trouvons des logiciels très semblables aux logiciels propriétaires en terme de processus de conception: pour des raisons d'ordre technologique, il a fallu un investissement humain et financier très conséquent pour le développement initial. Ensuite certaines raisons, dont le souci d'image ou la pérennisation du produit, ont poussé le

²voir le livre blanc de Smile « *Introduction à l'Open Source et au Logiciel Libre* » pour plus de détails sur l'esprit et les ressorts de l'open source

ERP open source

sponsor du moment à reverser le code en temps que vrai projet open source. Ce sont donc des projets assez matures qui ont été livrés à la communauté open source.

L'inconvénient est que la conception initiale elle-même a peu bénéficié de la collégialité propre aux projets open source, seules la maintenance et les évolutions ultérieures ont pu être inspirées par les têtes pensantes de la communauté open source.

Précisons néanmoins qu'à la différence des vrais logiciels propriétaires, leur licence leur a permis d'intégrer certaines briques open source de qualité dont le client final est soulagé.

Les archétypes de ces modèles de développement sont Compiere d'une part et Open For Business (Ofbiz) d'autre part.

La lignée Compiere

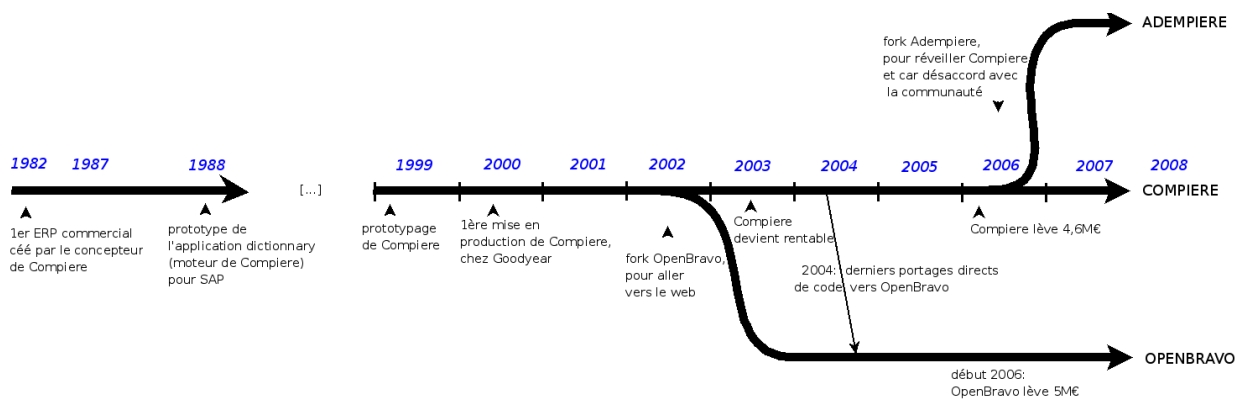


ILLUSTRATION 2: COMPIERE ET SES FORKS

Compiere est un projet développé à ses débuts par l'allemand Jorg Janke, lequel a su mettre à profit ses **20 années d'expérience chez SAP puis Oracle** et sa maîtrise des produits **ADV/Orga, Unisys, R/2 et R/3**. Les concepts de « l'application dictionary » (modèle de méta-programmation à la base de Compiere permettant l'adéquation de la persistance relationnelle avec les structures de données métiers personnalisées et leurs interfaces, voire page 67) ont été prototypés dès 1988 pour SAP en Smalltalk. Jorg Janke a ensuite mis ses idées de développement au service du projet libre Compiere dont le prototypage a commencé dès 1999, au début encore de l'ère Java.

La première société à avoir sponsorisé Compiere puis à l'avoir déployé en production est Goodyear Germany, en mai 2000. Compiere a rapidement connu des succès chez quelques grands comptes et grandes PME du secteur de la distribution et de la logistique. Ce succès est notamment imputable au support de la base de données Oracle, très présente sur ce segment, et à sa technologie 100% java, très mature.

ERP open source

Paradoxalement, ce développement spécifiquement pour Oracle a ensuite joué des tours à Compiere et ses forks. En effet, même si poussée dans ses derniers retranchements, PostgreSQL est légèrement moins performante, la base libre attire désormais énormément les petites PME et TPE pour des raisons évidentes d'économie de licence. Basé sur le périmètre de compatibilité défini par la norme SQL2003, le portage de la lignée des Compiere vers PostgreSQL a heureusement été achevé en 2006. En 2008, il est désormais éprouvé.

Mais l'histoire de Compiere est aussi celle de ses forks: en 2002, pour être indépendants, et parce que, contrairement à Compiere, ils pariaient déjà tout sur le web, une petite entreprise espagnole a créé Tecnicia qui deviendra ensuite **Openbravo**. Openbravo a forké tout le code métier (PL/SQL) de Compiere ainsi que son moteur de gestion d'entités (*l'application dictionary*) et l'a transposé dans une **interface web**. Depuis les deux bases de codes ont évolué séparément. En 2005, Openbravo a levé **5 Millions d'euros** et est parti à la conquête mondiale du marché des ERP pour PME sur le modèle d'un développement communautaire.

Compiere a lui aussi levé des fonds: **4,6 Millions d'euros en 2006**, là aussi pour prendre le plus possible de parts de marché aux ERP propriétaires. Mais au lieu comme Openbravo d'assumer un business model communautaire, les dirigeants de Compiere Inc se sont engagés sur un business model très proche des ERP propriétaires: à savoir une barrière d'entrée significativement élevée: toute documentation payante, pas ou très peu de développement communautaire, pas de consultation des contributeurs dans les process de décision, forums de discussion libre déserts, pas de wiki, pas de chat, pas d'ouverture vers les amateurs ou les développeurs sans moyen des pays émergents pourtant les plus intéressés par les ERP open source.

En fait Compiere a cherché peut-être à renflouer ses investisseurs trop directement alors qu'avec l'open source, à l'aune des succès de JBoss, MySQL ou Alfresco, il y a rarement retour sur investissement sans d'abord un investissement indirect clair dans le modèle communautaire (cf notre bilan sur Compiere page 93)

C'est cette logique de rentabilité trop directe et cette absence de bénéfice pour la communauté des développeurs originels qui a poussé ces derniers à créer en 2006 le fork **Adempiere**. Aujourd'hui, l'attentisme est de rigueur à l'égard d'Adempiere: il se pourrait bien qu'il en émerge un féroce concurrent de Compiere ou d'Openbravo, mais pour l'instant les seuls à tirer leur épingle du jeu sont les premiers intégrateurs à l'origine du fork. En effet, leur organisation globale reste incertaine et devra encore faire ses preuves pour aborder les défis que doit relever Adempiere: interface web, simplification de l'intégration, verticalisations...

Si Compiere, Adempiere et Openbravo atteignent aujourd'hui un ensemble fonctionnel honorable avec une tenue en charge éprouvée, leur futures évolutions devront affronter la grande inertie d'une base de code devenue lourde et peu factorisable. On reprochera notamment à ces plateformes d'être trop monolithiques et d'utiliser des paradigmes de programmation

ERP open source

peu productifs et dépassés comme le code fonctionnel intégralement en pl/SQL alors qu'on lui préférerait aujourd'hui une abstraction "objet".

Or c'est justement de ces abstractions « objet » qui masquent la complexité aux développements tierces et encapsulent les responsabilités que pourraient naître les verticalisations tant espérées. **Sans remise en cause technique, la lignée des Compiere est moins adaptée à la catalyse d'un développement communautaire de modules verticaux, car l'actuelle unité de code réutilisable, la procédure stockée, n'a pas une granularité assez fine** et donc les développements d'un projet particulier sont moins souvent reversés à la communauté pour servir les projets suivants. A l'inverse une solution comme TinyERP encourage davantage la réutilisation et a donc pu bâtir une bibliothèque de plus de 200 modules fonctionnels réutilisables et combinables à souhait.

Aujourd'hui Openbravo est la lignée de cette espèce qui se distingue par une forte volonté et capacité de conquête et on peut donc en attendre une forte dynamique de développement (cf notre bilan sur Openbravo page 98). Il est aussi important de noter que derrière leurs vitrines reluisantes, **beaucoup d'ERP propriétaires sont largement moins bien conçus**. Enfin, lorsqu'il a créé Compiere, le concepteur de Compiere connaissait tous les secrets architecturaux des alternatives commerciales des gros ERP tels que SAP R/3 qui dominent aujourd'hui encore le paysage ERP.

Dernière minute: ce 19 Mai 2008; Openbravo vient de lever à nouveau 12 Millions de dollars de la part du groupe d'investissement européen Amadeus! Cela confirme bien la montée en puissance de cette lignée issue de Compiere et rend totalement crédible les plans annoncés en terme de perfectionnement de la technologie, d'extension fonctionnelle et de localisation. Par ailleurs Openbravo continue à afficher sans ambiguïté son ancrage open source et l'investissement dans sa communauté.

la lignée Ofbiz

ERP open source

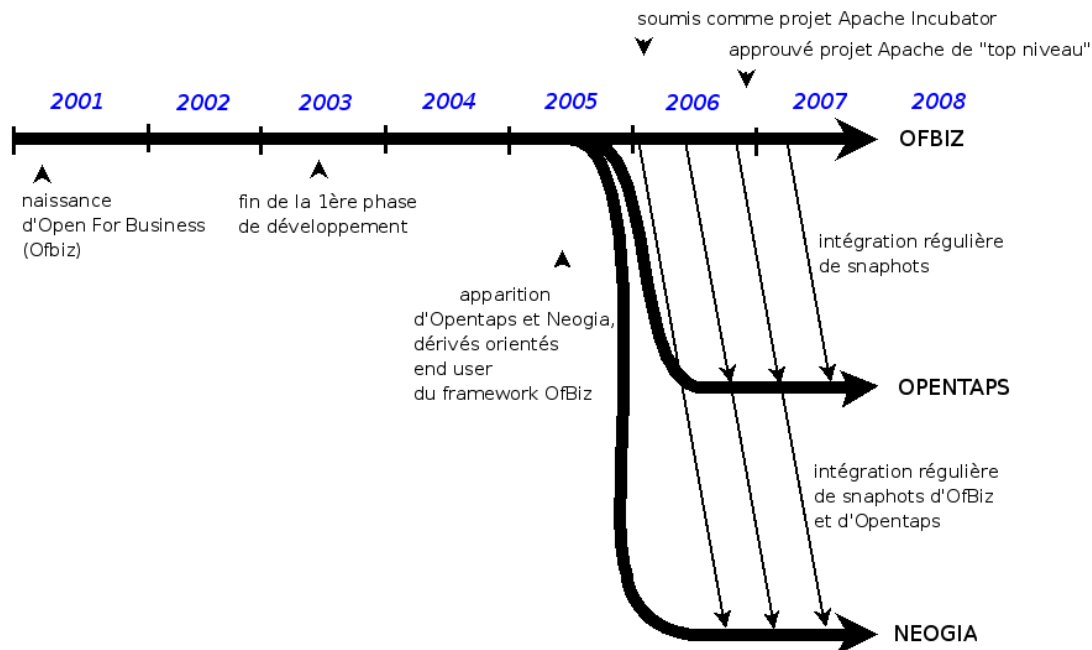


ILLUSTRATION 3: OFBIZ ET SES DÉRIVÉS

Le projet Open For Business (« Ofbiz ») est né en 2001 et a terminé sa première phase de développement vers 2003 sous l'impulsion de ses 2 créateurs: David Jones et Andrew Zenesky. Depuis le projet s'est enrichi de nombreux modules fonctionnels. En 2006 il a même été soumis comme projet "incubator" à la fondation **Apache**. Il en est très rapidement sorti comme projet approuvé de top niveau.

Ofbiz est un peu différent de Compiere. En effet, si Open For Business est bâti sur les même concepts que Compiere, modélisation par entités plutôt qu'objet et interface méta-programmée, le projet n'a pas ce volume de code PL/SQL, peu propice à l'extensibilité. En ce sens il a de meilleures bases techniques, bien qu'il ait d'autres défauts.

Cependant, à cause de cette modélisation par entités génériques et non par objets métiers et de ses racines dans un langage à dominante statique, Java, qui tend à paralyser les cycles de développement par rapport à des langages dynamiques, l'ensemble global du projet Ofbiz reste, à l'instar de Compiere, un gros projet qui a représenté un énorme investissement et qui évoluera désormais lentement, alourdi de son fardeau historique et de ses choix de modélisation sous-jacents discutables.

Si Ofbiz a, par certains aspects, une meilleure qualité technique que la lignée Compiere, **il a pourtant moins percé dans le monde des progiciels**. Cela vient sans doute du fait qu'Ofbiz a eu la prétention plus modeste de n'être **qu'un framework** adapté pour développer des logiciels d'entreprise **et non un ERP complet clé en main** soutenu par un éditeur fort. Le produit s'est notamment spécialisé dans l'interface d'e-commerce sur le modèle de ce qui se faisait dans les années 2003.

ERP open source

Si Ofbiz fut sans conteste un framework utile dans le monde J2EE pour construire de grosses applications de vente en ligne, on peut néanmoins légitimement se poser la question de sa compétitivité actuelle face à des frameworks plus modernes.

Par ailleurs, dans la mesure où l'application d'e-commerce recherchée serait au contraire plutôt standard, il pourrait être au contraire plus judicieux de brancher une solution d'e-commerce plus spécialisée et plus compétitive (comme Magento) avec un ERP plus mature et moins cher à intégrer.

Enfin, si Ofbiz lui-même n'est donc pas à recommander pour un ERP clés en main, il n'en reste pas moins que deux projets d'ERP très sérieux se servent d'Ofbiz comme base et y apportent le fonctionnel complémentaire et le packaging pour constituer cette fois de véritables ERP: il s'agit d'**Opentaps** d'une part et de **Neogia** d'autre part. Neogia importe même le fonctionnel CRM d'OpenTaps. A noter que Neogia, outre ses apports fonctionnels et sa francisation, apporte aussi une rustine technique aux défauts de conceptuels d'Ofbiz: il permet de générer par modélisation **UML** des objets métiers ayant des comportements et pas seulement des états. Ces méta-modèles génèrent ensuite les entités Ofbiz de bas niveau correspondantes. Si ceci paraît séduisant sur le papier, nous verrons qu'il convient de rester prudent avec Neogia en particulier à cause du manque de masse critique de sa communauté (cf notre bilan sur Neogia page 109).

b) Les outsiders 'low cost', nés de ruptures technologiques

A peine quelques années plus tard, on a vu apparaître une nouvelle classe d'ERP dont les fondations conceptuelles plus ambitieuses ont rendu plus simple le développement des solutions métiers, lesquelles restaient néanmoins fortement inspirées des acquis fonctionnels de Compiere ou d'Ofbiz.

Profitant de ces nouvelles plateformes conceptuelles encore inégalées dans le monde des ERP commerciaux, cette seconde classe d'ERP a pu se développer avec des budgets sensiblement moindres et ont été **auto-financés au fur et à mesure de leurs premières mises en oeuvre**. Aujourd'hui, leur reconnaissance sur le marché n'est liée qu'à leurs qualités donc et non à l'efficacité de leur marketing.

On trouve dans cette seconde catégorie d'ERP **TinyERP (rebaptisé OpenERP depuis le 25 Avril 2008) et ERP5**. (cf notre bilan sur TinyERP page 104 et sur ERP5 page 96)

Ce qui change fondamentalement dans la conception de TinyERP et ERP5, c'est qu'ils disposent d'une **modélisation** qui suit fidèlement les préceptes de la programmation dite « objet ». En conséquence, le fonctionnel métier peut y être surchargé au gré des besoins de verticalisation pour atteindre exactement les fonctionnalités souhaités sans pour autant engendrer de code cher à maintenir ou à intégrer, *les fameuses usines à gaz*.

ERP open source

Plus précisément, la capacité **d'héritage** de la modélisation dite *objet* permet de définir les concepts et comportements métiers par surcharge de modèles pré-existants, c'est à dire en ne spécifiant que les écarts à ces étalons au lieu de devoir redéfinir chaque modèle en partant de zéro ou encore d'embarquer systématiquement d'office tous les paramétrages imaginables ce qui mène à des intégration trop coûteuses pour les PME et TPE. Par exemple, un produit pour l'agro-alimentaire dont on devra considérer la date de péremption, dans le calcul du prix de vente par exemple, ne sera qu'une spécialisation opportuniste du modèle générique de produit. A charge à l'entreprise de l'agro-alimentaire d'installer ce plugin alors que, par exemple, l'entreprise de service, elle, ne se verra pas ses coûts d'intégration s'envoler avec des paramétrages inutiles à son secteur d'activité. L'unité de code réutilisable dans l'écosystème open source, est fine et permet à plus de gens de fédérer leur développements spécifiques en des **plugins** cohérents et réutilisables directement.

D'autre part, la modélisation dite objet va apporter à ces ERP des propriétés dite **d'encapsulation**: c'est à dire qu'on délègue explicitement certaines responsabilités à certains concepts (ou objets). Ce qui permet: d'apporter de la lisibilité au code et de le stabiliser en évitant les effets de bords. Par exemple, le calcul du prix d'un produit est localisé un à un endroit précis unique et dont le langage lui même contrôle l'accès afin que tous les développeurs, y compris les tiers intégrateurs, puissent facilement comprendre toute fonctionnalité mettant un jeu le prix de vente d'un produits et garder la maîtrise lors des nécessaires personnalisations.

Les lecteurs renseignés remarqueront que l'on parle ici d'ERP écrits en langage Python alors que ceux de la première classe le sont en langage Java. Pourtant ces deux langages sont bien des langages permettant la modélisation objet. Mais il se trouve que les seuls ERP fonctionnellement matures en langage Java n'utilisent pas encore la modélisation dite objet, ou alors assez imparfaitement comme sur Neogia tout au plus.

L'explication est que Java est un langage appartenant à la famille des langages de typage dit *fort et statique*: c'est à dire qu'on doit déclarer, avant même la phase de compilation et le déploiement, les classes de comportements attendus des entités de modélisation. Or, techniquement parlant, ceci cadre très mal avec la possibilité de personnalisation à chaud de ces modèles de comportement. Et pourtant c'est bien ce qui est recherché dans l'ERP: un paramétrage assez poussé sans avoir à rien recompiler ni à redéployer: **idéalement, le plus de paramétrage possible sera assumé par des consultants métiers sans avoir à recourir à des développements coûteux**. C'est précisément ce qui était visé par l'*application dictionary* de Compiere également repris par Openbravo: permettre l'altération à chaud des modèles de données et de leurs écrans. Or, étant basé sur un langage de typage statique fort, la famille des Compiere a dû renoncé à la *modélisation objet* pour permettre une certaine dose de paramétrage à chaud (celle des structures de données). Au contraire, **les ERP basé sur Python, un langage typé fort dynamiquement on n'ont pas eu à renoncer à la modélisation objet**

ERP open source

pour permettre un maximum de capacités de personnalisations à chaud par des non développeurs. Quant aux ERP de la famille d'Ofbiz/Neogia, ils embarquent une grande dose de modélisation objet mais sacrifient eux au paramétrage à chaud par des non-développeurs.

Un corollaire important de ce bond conceptuel réalisé par ces nouvelles plateformes ERP est le coût sensiblement plus faible pour qu'une tierce partie développe un module d'extension. Ces ERP ont été développés à moindre frais, et ils connaissent aujourd'hui une "**catalyse**" open source plus importante dans la mesure où plus de modules ont tendance à être contribués par des tierces parties et ces modules ont tendance à être ensuite plus maintenus et réutilisés à bon escient (notamment par d'autres modules justement). C'est ainsi toute une galaxie de composants fonctionnels qui s'échafaude naturellement autour de ces catalyseurs.

Attention néanmoins, si l'aspect catalytique du développement communautaire est vrai pour TinyERP, il ne l'est pas pour ERP5 qui rebute beaucoup par son système de persistance exotique (cf page 64) d'une part et son business model très centré autour de l'éditeur d'autre part.

Quand peut-on envisager un ERP open source?

Par ce qu'ils ne résultent que de la mutualisation d'efforts de développements sur des problématiques génériques et parce qu'ils ne peuvent pas encore bénéficier de budgets R&D aussi conséquents que leurs équivalents propriétaires, **les ERP libres ne sont pas adaptés à tous les projets.**

Ainsi, un ERP libre ne sera pas compétitif sur un secteur historique très spécifique, par exemple un ERP pour la fabrication automobile, là où des éditeurs traitent le problème depuis longtemps.

En effet, si l'ouverture du code est toujours souhaitable, l'assiette des clients qui financent un développement très spécifique ne change pas fondamentalement entre un logiciel open source et un autre propriétaire.

Au contraire, pour un projet si spécifique qu'aucun ERP propriétaire du marché n'est adapté, se baser sur un ERP libre bien conçu sera un moindre mal. De même, sur un périmètre générique natif en général restreint (par exemple en utilisant une comptabilité générale externe), un ERP libre pourra représenter une économie.

On peut résumer avec le tableau suivant:

ERP open source

Cas où on peut envisager un ERP open source	Cas où un ERP open source risque de ne pas être adapté
<p>Compétitivité native: Les fonctionnalités recherchées conviennent sans aucun développement dans l'ERP libre, généralement parce qu'on envisage un périmètre restreint.</p> <p>Compétitivité par la souplesse: besoin, au contraire, de concevoir un véritable mouton à 5 pattes pour répondre aux besoins d'une activité très spécifique et non traitée par les ERP abordables du marché. Bâtir autour d'un ERP open source sera alors moins risqué et plus économique.</p> <p>Croissance incompatible avec le système de tarification des ERP commerciaux: entreprise en forte croissance qui n'avait pas encore d'ERP. Vous désirez aller plus loin qu'une simple gestion comptable et ne pas avoir de coût d'ERP liés à votre croissance.</p>	<p>Grand compte: il est probable qu'un budget très conséquent vous donne accès à des ERP propriétaires plus adaptés.</p> <p>Incertitude sur la tenue en charge: si vous dépassez plusieurs centaines de milliers de transactions par mois et/ou plusieurs dizaines d'utilisateurs faisant des requêtes simultanées, alors il faudra se poser des questions et éventuellement faire des tests de charge au préalable. Bien sûr, la capacité de montée en charge augmentera avec le recul.</p> <p>Il existe des ERP propriétaires abordables très spécialisés pour votre secteur d'activité. Notez que la souplesse des moteurs de workflow proposés par certains ERP cités permet une grande adaptabilité, néanmoins, si les traitements métiers sont très spécifiques, les ERP du marché seront vraisemblablement plus appropriés.</p>

ERP open source

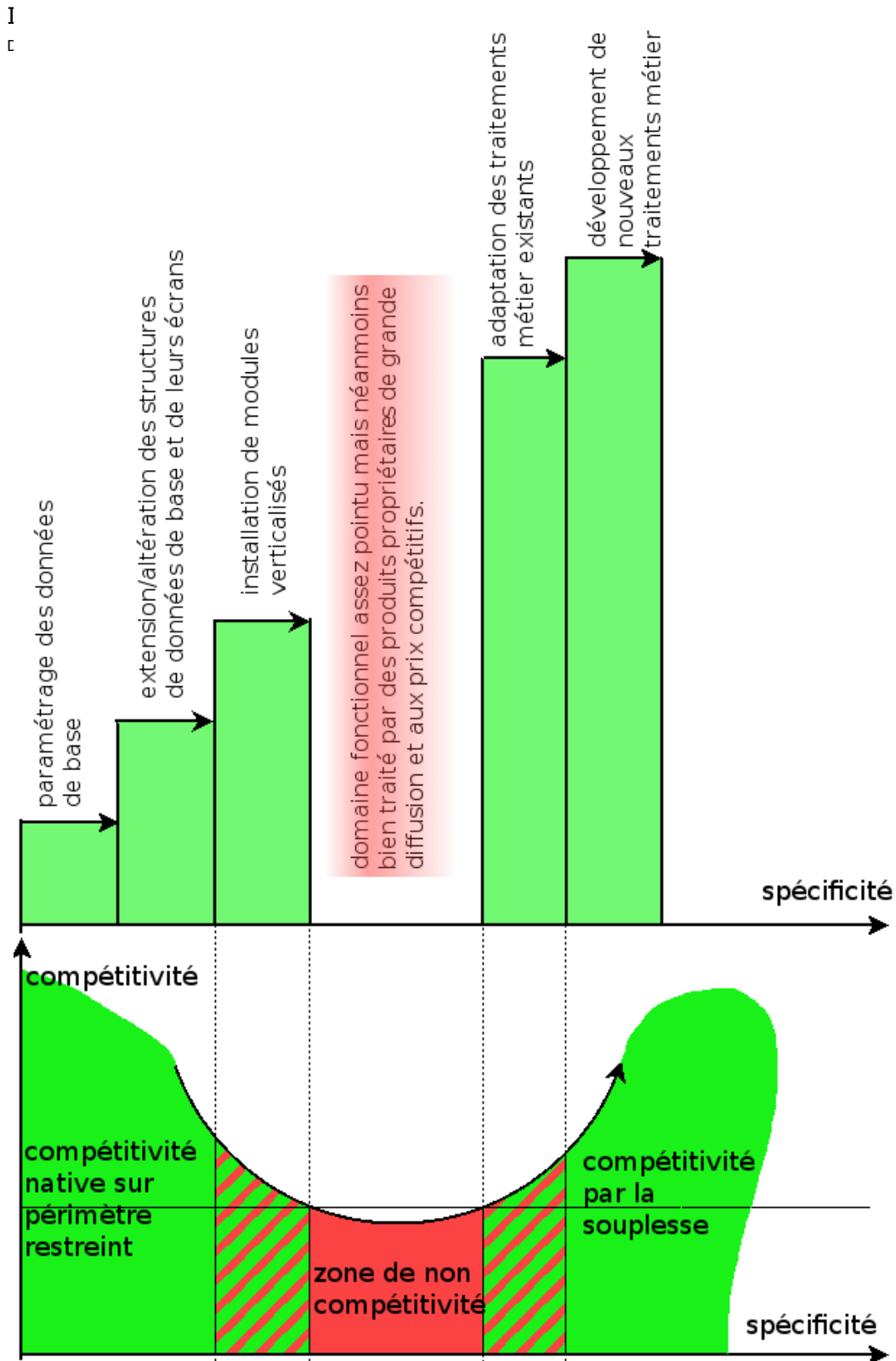


ILLUSTRATION 5: COMPÉTITIVITÉ SELON LA SPÉCIFICITÉ DE L'ERP, EN FONCTION DES PRINCIPAUX MODES DE PERSONNALISATION, PAR COMPLEXITÉ CROISSANTE; BIEN SÛR, LE DOMAINE DE NON COMPÉTITIVITÉ, DÛ PAR EXEMPLE AU FARDEAU DE LA COMPTABILITÉ GÉNÉRALE, TEND À S'ESTOMPER.

ERP open source

Quels produits commerciaux sont concurrencés par les ERP open source présentés?

Les ERP abordés dépassent mais ne remplacent pas toujours les logiciels communs de la gestion comptable, à savoir:

- Ciel
- La gamme Sage

Il est courant qu'un déploiement d'ERP s'intègre au progiciel de comptabilité au moyen d'un *pont comptable*, c'est-à-dire un échange d'écritures.

Sans prétendre à l'exhaustivité, les ERP abordés concurrencent potentiellement les ERP propriétaires suivant:

- SAP Business One
- SAP Business ByDesign
- SAP All in One
- Sage X3
- Sage Adonix
- Microsoft Dynamics / Navision
- gamme Cegid
- Baan
- Peoplesoft
- Divalto
- Silog
- Prism

Les ERP qui ne sont à priori peu concurrencés par les ERP open source:

- SAP R/3
- Oracle Applications
- Siebel (désormais partie CRM d'Oracle).
- Lawson

D'une manière générale, nos ERP libres ne sont pas encore à la hauteur en terme de planification automatique et optimisée de la production en tenant compte des contraintes sur les ressources ni dans des secteurs très spécifiques déjà bien traités par les ERP commerciaux, par exemple les capacités de profilage natives des clients en CRM dans Siebel.

Néanmoins, même pour ces ERP haut de gamme, il existe des cas d'intégration où seule une partie des fonctionnalités sont recherchées et

ERP open source

où au contraire on tient à implémenter des règles métier spécifiques. Dans ce cas là, les ERP libres peuvent malgré tout concurrencer R/3, Oracle ou Lawson.

De plus, même s'ils pourraient être adaptés, les ERP libres n'ont pas encore prouvé leur tenue en charge dans des environnements de sollicitations extrêmes, là où les SAP R/3 ou Oracle Applications ont eux déjà fait leurs preuves.

Enfin, les plus gros ERP cités sont nativement interfaçables avec une foule de modules propriétaires spécialisés, par exemple de planification, de prévision des ventes, de messages EDI, de configurateurs de produits, de gestion de stocks partagée avec fournisseurs et clients alors que les ERP libres n'ont pas encore d'interfaces natives avec ces progiciels.

Finalement, un bon moyen de savoir quand un ERP libre est concurrentiel est le raisonnement suivant: si un Microsoft Navision ou un SAP Business One semble à peu près dimensionné pour votre entreprise mais qu'il faudrait y intégrer des développements spécifiques, alors un ERP libre devrait vous permettre des économies importantes.

LES DOMAINES FONCTIONNELS FONDAMENTAUX DES ERP

Nous passons en revue dans cette partie les grandes fonctionnalités d'un ERP généraliste, en illustrant nos propos par des exemples concrets sur les différents ERP retenus.

Paramétrage de base, produits et partenaires

Dans son usage le plus dépouillé, un ERP sert à gérer les transactions de différents produits entre différentes entités commerciales dans le but de permettre de produire les pièces comptables légales d'une part et d'optimiser les résultats de l'entreprise par un meilleur pilotage d'autre part. C'est notamment l'usage essentiel qu'attend le secteur de la distribution. Cet usage met déjà en jeu les entités fondamentales de l'ERP que nous allons présenter.

Produits

Les structures de données les plus élémentaires d'un ERP sont les partenaires commerciaux et les produits. Par souci de généricité, on parle de *produits* au sens large, en incluant aussi bien les **biens de distribution**, que les **produits fabriqués** ou les **services**. Un paramétrage fin, éventuellement pré-packagé dans un module verticalisé, va différencier les comportements de ces produits. Un service sera par exemple rattaché en tant que tâche d'un projet alors qu'un produit physique déclenchera des approvisionnements ou des ordres de fabrication.

Un produit est donc une entité, physique ou non, qui donne lieu à une facturation comptable et dont on désire tracer les transactions. Pour éviter que les produits ne soient trop complexes à créer à chaque fois, tant Openbravo que TinyERP par exemple, offrent des options de création à partir d'un **template** ou encore avec des **champs remplis par défaut** et dépendant éventuellement d'autres champs (TinyERP). On peut également créer un catalogue à partir de données tabulées.

Par ailleurs, la famille des Compiere, TinyERP, ERP5 et OfBiz/Neogia intègrent un système de **produits à variantes**: c'est-à-dire une famille de produits semblables en tous points, exceptés quelques paramètres isolés qui peuvent être localement surchargés, par exemple la couleur sur d'un T-Shirt. Par défaut, ces produits sont soumis aux mêmes règles de prix et de production. On peut bien sûr tenir compte de la variante pour

ERP open source

surcharger l'une des règles de prix ou de production, mais l'intérêt est qu'on n'y est pas contraint et qu'on peut ainsi directement hériter les comportements attendus.

En revanche, **aucun des ERP que nous avons abordé ne possède de configurateur de produit**. C'est ce qui est mis en jeu par exemple lorsqu'on achète un ordinateur chez Dell: on choisit toutes les options du produit. C'est une problématique qui peut être complexe, car il y a des dépendances entre les options: par exemple, si on prend une carte mère bas de gamme, alors impossible d'y monter un processeur trop puissant. De même si on ne choisit pas de doubler la mémoire RAM, alors peut être qu'on pourra avoir une promotion sur l'achat d'une imprimante...

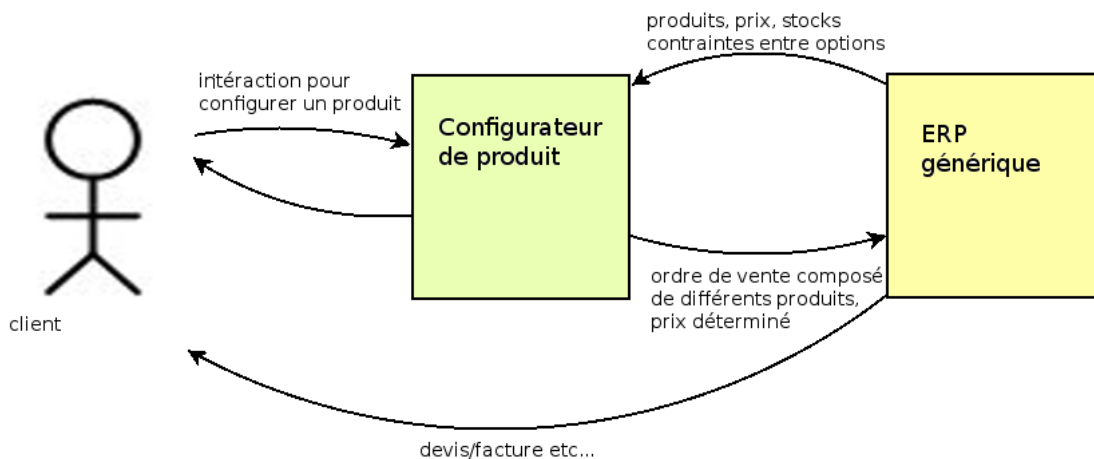


ILLUSTRATION 6: COMMENT AJOUTER UN CONFIGURATEUR DE PRODUIT SUR UN ERP OPEN SOURCE

Si ce système n'est pas natif, on peut imaginer qu'une interface, web d'e-commerce par exemple, soit spécifiquement développée pour gérer les dépendances entre options, techniquement parlant les variantes multiples, et promotions et qu'au final elle ne fasse que transmettre à l'ERP un ensemble de produit à acheter associés à un *package* et dont le prix est fixé. Notons qu'il n'est en général pas possible de créer autant de modèles de produits qu'il y a de combinaisons d'options car leur nombre ainsi que la complexité de la gestion des dépendances exploserait. De même, l'approvisionnement de produits configurés demandera des développements similaires.

Partenaires commerciaux

De même, les *partenaires* sont des entités génériques qui incluent la société elle-même, ses clients, ses fournisseurs et ses sous-traitants. Dans TinyERP, les partenaires peuvent être issus d'un annuaire LDAP nativement, et sur Openbravo un développement en ce sens est possible. Tant TinyERP qu'Openbravo permettent de modéliser la structure éventuellement **hiérarchique** d'un partenaire commercial (filiales, directions, divisions, etc.).

ERP open source

Par ailleurs les partenaires peuvent avoir plusieurs personnes comme *contact* et plusieurs adresses. Ils ont aussi un pays et une devise propre, ce qui sera prise en compte lors de l'émission des factures et autres pièces comptables.

Paramétrage comptable

Enfin, pour permettre l'intégration comptable des transactions de produits entre partenaires, l'ERP permet aussi le paramétrage des plans comptables de comptabilité générale et analytique, des devises à utiliser ou encore des unités de mesure de produits (unité, longueur, poids...). Lorsque la comptabilité devra être exportée vers un outil dédié, il faudra en premier lieu importer dans l'ERP la structure des journaux et plans de comptes pour que l'ERP soit en phase avec le logiciel comptable cible.

Pricing

Par défaut, les produits ont un prix unitaire propre, lequel peut-être un simple prix catalogue (méthode du prix « standard »), ou au contraire un prix dérivé des coûts de revient passés (par exemple la méthode du « prix de revient moyen » majoré d'une marge définie) dans la mesure où l'on souhaite maîtriser la production et où l'ERP le permette.

L'**unité**, justement peut être définie arbitrairement: pièce, poids en Kg, en tonne, mètre, litre... L'essentiel est qu'il existe des facteurs de conversion entre unités si elles diffèrent entre prix unitaire et unité de commande par exemple.

De même, on peut changer de **devise** à loisir à condition qu'il existe un taux de conversion entre devises pour la période en cours. Pour les devises, ces taux de conversions peuvent être différents selon le sens de la conversion.

A partir de ce prix unitaire par défaut, un prix réel sera attribué lors d'une transaction de vente. Lorsque toutes les conditions d'une transaction commerciale sont connues, comme lors de l'émission d'un devis, le prix réel appliqué sera alors soit surchargé à la main, soit calculé à partir du prix par défaut auquel on appliquera des règles de prix, lesquelles tiennent au moins compte du partenaire ou encore de la quantité vendue. La famille des Compiere, ainsi que TinyERP, offrent ainsi un **moteur de règle de prix** fonctionnellement similaire. Neogia embarque lui le moteur de règles d'OfBiz qui tient compte également des catégories des produits.

Dans le cas d'une intégration avec une plateforme d'e-commerce par exemple, on pourra soit s'en remettre à l'ERP pour fixer le prix d'un produit, soit laisser l'outil d'e-commerce utiliser par exemple son propre moteur de promotion pour fixer un prix, puis transmettre la transaction à l'ERP.

ERP open source

Par ailleurs, il pourra être intéressant de personnaliser l'ERP pour y intégrer différentes **politiques commerciales**. Par exemple selon la probabilité de disponibilité d'un produit à échéance, afin d'optimiser les ventes et de diminuer les stocks. Ceci est d'autant plus pertinent lorsqu'on veut travailler en flux tendu ou optimiser la vente de produits périssables (agro-alimentaire, transport, événementiel par exemple). Ainsi, l'achat d'un produit avec une certitude de disponibilité de 100% à la date t coûtera sensiblement plus cher que la certitude de 95% à cette date t. Ne vendre que les produits disponibles avec 100% de certitude sur un délai est souvent non optimal.

Par exemple pour le trafic aérien, il est bien plus intéressant d'arriver à des taux de remplissage des avions (un place est une denrée périssable) de 90% en pratiquant le surbooking. Sur un vol il est très probable que certains clients soient alors ravis de 'sacrifier' leur place en contrepartie d'une généreuse indemnisation, laquelle reviendra moins cher à la compagnie qu'un taux de remplissage moindre de 80%. De même un client de 1ère classe ne sera pas sujet aux aléas du surbooking alors qu'un client de 2nde admettra de prendre ce faible risque. Il est intéressant de vendre, même à prix cassé, une place de spectacle restante qui va débiter dans quelques heures seulement. Dans l'agro-alimentaire, il peut encore être intéressant d'inciter la vente de produit qui risquerait d'arriver à péremption sinon.

L'ERP open source est donc la plateforme idéale sur laquelle ces politiques de ventes et de tarifications peuvent être implémentées. Les règles de pricing intégrées nativement dans les ERP étudiés sont néanmoins plus basiques: **par défaut elles ne savent tenir compte que du client, du pays, du produit ou de sa catégorie et des quantités**. Typiquement, sur la famille de Compiere ou sur TinyERP, on donne différentes règles d'altération du prix par défaut selon ces paramètres et avec un ordre de priorité, dès qu'une règle s'applique, le prix est déterminé.

On peut, bien sûr, sortir de ce comportement par défaut et par exemple tenir compte de champs personnels pour les règles à appliquer. Si l'ajout dans les interfaces et le stockage de ces champs personnalisés est une action assez simple, tenir compte de ces nouveaux champs demande en revanche de faire appel à des développeurs plus ou moins spécialisés selon la technologie choisie. Par exemple il faudra quelqu'un de compétent en PL/SQL (Oracle) ou PG/SQL (PostgreSQL) pour faire ce développement sur la famille de Compiere, une compétence moyenne en Python sera nécessaire pour le réaliser sur TinyERP.

Ventes et gestion des partenaires (CRM)


Centralisation et historisation des données client

En plus de gérer la vente de produits, l'ERP centralise les données en temps réel sur les clients, les stocks et la politique tarifaire de l'entreprise. Ainsi, on est bien sûr d'**annoncer des délais corrects et d'appliquer les promotions conformes à la stratégie d'entreprise et au profil du client.**

L'ERP permet aussi d'améliorer la relation client en simplifiant la gestion de ses paiements. Diverses règles de paiement peuvent être paramétrées: paiement à la commande, paiement à la réception avec échéance... La gestion centralisée des comptes clients permet aussi la vente de packages composés de produits ou services que le client pourra choisir dans les limites d'un forfait: tant que son crédit est positif, les achats seront automatiques et un nouveau paiement ne sera demandé que lorsque le crédit est épuisé.

La centralisation des données clients et l'automatisation des émissions de devis, factures, relances, expéditions, paiements et reçus **évite aussi les erreurs de saisie** sur les montants et références des clients et permet bien sûr de réduire les délais de traitement. Sur Openbravo ou TinyERP, cette centralisation est facilitée par la possibilité d'attacher des items entre eux: par exemple une facture spécifique avec un produit spécifique. Les outils de reporting permettent alors en un clic de retrouver l'historique des ventes et autres relations rattachées à un client précis.

ERP open source

powered by **openbravo** Carte de tiers 

Information générale

DONNEES GENERALES	Alias produit: sailor Nom commercial: Sailor Inc Nom Fiscal: Sailor Inc TIN/Id: Groupe tiers: Customers Date inscription: 23-03-2006 Mettre à jour: 01-03-2007 Commercial: Non Employé: Non Actif: Oui	DONNEES GENERALES	Alias produit: Redbunny Nom commercial: Red Bunny Nom Fiscal: TIN/Id: Groupe tiers: Customers Date inscription: 31-03-2006 Mettre à jour: 01-03-2007 Commercial: Non Employé: Non Actif: Oui	Comme client	Type Fact: Después de entregado Calendrier Fact: Monthly 1st Règles Paiement: A la vista Période: One month Commercial: John Moneymaker Liste prix: General Sales A. Compte1: 43000 - Clientes de dudoso cobro A. Compte2: 43100 - Efectos comerciales en cartera A. Compte3: 43000 - Clientes (euros) A. Compte4: 43000 - Clientes (euros)
--------------------------	---	--------------------------	---	---------------------	--

Analyse cliente

Commandes jusqu'à 09-04-2008	2008	0	2007	0	0	.../0 %	
Expéditions 09-04-2008	2008	0	2007	0	0	.../0 %	
Factures à 09-04-2008	2008	0	2007	0	0	.../0 %	
	2008	0	Total 0	ABC PRODUCTS	2007	0	Total 0

Adresse

Adresse	Code Postal	Ville	Province	Pays	Téléphone	Téléphone 2	Fax	Livrai	Inv	Loc. I
Avenida Espinaca 14		Albarracín	TERUEL	Spain				Si	Si	No
Carrot street	44578	Carrot	HUELVA	Spain				Si	Si	No

Contactes

Nom	Email carte de crédit	Téléphone	Téléphone 2	Fax	Adresse
					Avenida Espinaca 14, Albarracín, TERUEL (Spain)

Autres

Banque	No. Compte
Uncle Scrooge	2102458758967789

Séq	Nom	%	Casc
10	Discount 03%	3	SI
20	Discount 07%	7	SI

ILLUSTRATION 7: CRM SUR OPENBRAVO: FICHE DES HISTORIQUES DE VENTE D'UN CLIENT

Affectation des clients et avants ventes aux commerciaux

Une fonctionnalité également très recherchée sur la CRM est l'affectation des clients ou prospects à des commerciaux en particuliers.

Ceci est particulièrement abouti sur Neogia et TinyERP. Ces ERP possèdent même des notions de tableau de bord personnalisés. Ainsi, dès qu'un commercial se loggue sur l'interface web de l'ERP, il accède d'un clic à sa liste de clients, ses cas à suivre et l'état de ses prospects. Dans TinyERP, les fonctionnalités avancées de calendrier seront mises à profit pour la planification des réunions et autres évènements commerciaux.

ERP open source

The screenshot displays the OpenERP CRM interface. At the top, there's a navigation bar with 'OpenERP' logo, 'Open Source Management Software', and user information: 'Bienvenue Administrator', 'Accueil', 'Préférences', 'A propos', 'Déconnexion'. Below this is a menu bar with 'MENU PRINCIPAL', 'RACCOURCIS', and various modules like 'Partners', 'My Meetings', 'My Leads', 'My Jobs Requests', 'Partners', 'Open Opportunities'. The main content area is titled 'CRM Dashboard' and contains several sections:

- My Leads:** A table with columns: ID du cas, Name, Phone, Email, Status, Lead Source, Campaign Type, Responsable, Statut. It lists 4 leads with details like 'Wendi Baltz', 'Diego Barga', 'Faustino Speegle', and 'Jesse Denning'.
- My Meetings:** A table with columns: ID du cas, Titre, Next Meeting, Section du cas, Priorité, Stage, Catégorie, Type, Responsable, Statut. It lists 4 meetings like 'Discuss pricing', 'Update the data', 'Initial discussion', and 'Follow-up on proposal'.
- My Cases:** A table with columns: ID du cas, Date, Deadline, Description cas, Partenaire, Responsable, Priorité, Revenus prévus, Coûts prévus, Statut. It lists 4 cases like 'Discuss pricing', 'Patches Error in Program', 'Disk failure', and 'Salesman'.
- My Leads By Stage:** A pie chart showing the distribution of leads across stages: New, Converted, and Assigned.
- My Sales Pipeline:** A pie chart showing the distribution of sales pipeline items: open, done, and draft.

ILLUSTRATION 8: PANNEAU PERSONNALISÉ FAÇON CRM POUR LES COMMERCIAUX DANS OPENERP (TINYERP)

Routage des mails et des appels téléphoniques

Dans TinyERP exclusivement, la notion de cas de suivi utilisée pour la CRM peut être facilement interfacée avec un la '**mail gateway**', un composant TinyERP qui permet d'intercepter les mails entrant et sortant et de les dispatcher sur les différents cas (et donc de les affecter) automatiquement selon les règles entrées dans le moteur de règles de TinyERP.

Un autre module nécessitant encore quelques améliorations de finitions permet d'intercepter les appels d'un call-center sur VoIP avec serveur **Asterisk** pour ouvrir les fiches clients correspondantes lorsque ceux-ci appellent. L'intégration avec un serveur de VoIP nécessitera néanmoins quelques développements et ne permettra pas d'atteindre le niveau de service d'un CRM Siebel. Notons aussi que pour ce dernier module, des développements raisonnables pourraient être réalisés pour implémenter la fonctionnalité sur les autres ERP ne possédant pas nativement la fonctionnalité.

A noter qu'un dernier module tout juste publié par l'intégrateur Heonium permet lui d'envoyer des faxes.

ERP open source

Campagnes marketing

Les ERP abordés ne vont ici pas aussi loin que les outils spécifiques tels que **SugarCRM** ou son fork **VTiger** qui sont spécialisés dans la gestion de campagnes multi-canal.

Cependant, dans TinyERP, un module permet l'envoi de **SMS** via un opérateur externe et du **mailing** de masse. Étant donnée la souplesse de cet ERP, il faudra se poser la question de savoir si ces fonctionnalités natives, éventuellement agrémentées de quelques développements spécifiques ne pourraient pas vous suffire.

Ciblage des clients

Le ciblage, ou encore profilage, permet de ranger les clients dans des catégories selon certaines règles plus ou moins immédiates pour ensuite entreprendre des actions commerciales ciblées à leur égard. Le but étant de fidéliser des clients ou encore d'essayer d'augmenter les marges sur des profils à plus fort potentiel.

Cependant, les ERP abordés ne sont pas ici aussi complets que les ERP propriétaires spécialisés. Au mieux, on peut, en général, utiliser les fonctions de reporting natives de l'ERP pour déduire les catégories de clients. TinyERP va lui un peu plus loin avec un module de profiling qui classe les clients, en leur assignant une ou plusieurs catégories selon des règles qu'on entre dans le moteur de règles.

Pour aller plus loin dans le profiling, toujours dans l'open source, on pourra utilement soumettre le référentiel des clients avec leur historique des ventes à des outils de Data Mining tels que **Weka** (solution rachetée par Pentaho) pour **inférer des règles** de corrélations entre données. Par exemple la déduction d'une règle comme: *les clients de telle région ne tendent à acheter que des offres sur lesquelles les marges sont faibles et ce d'autant plus que le produit est cher*. Mais il est possible aussi de classer plus efficacement les profils avec des méthodes de **clustering** telles que les nuées dynamiques.

Enfin, si l'ERP ne suffit pas ou si les développements complémentaires sont trop conséquents, on préférera s'interfacer avec un progiciel de CRM spécifique, notamment SugarCRM, son fork VTiger ou encore Concurive CRM (ex Centric). On pourra aussi mettre à profits la panoplie d'ETL libres aboutis tels que **Kettle** de Pentaho ou **Talend** pour construire rapidement et de façon maintenable des ponts entre les logiciels. Cette option est spécialement conseillée pour du CRM haut de gamme avec Openbravo qui nativement en inclut moins que Compiere. Il faut savoir que l'éditeur de Openbravo travaille au développement d'un connecteur pour SugarCRM et prévoit aussi d'enrichir le périmètre natif d'Openbravo sur la CRM.

Vente par plateforme e-commerce

Neogia est le seul ERP à embarquer nativement une plateforme d'e-commerce avec gestion du catalogue sur un frontal de vente, gestion du panier d'achat et des paiements. Il faut cependant noter que cette capacité, héritée du framework OfBiz, si elle a le mérite de tenir de très fortes charges pour du commerce B2C, possède une ergonomie et une souplesse assez éloignée des meilleurs frontaux d'e-commerce spécialisés tels que Magento.

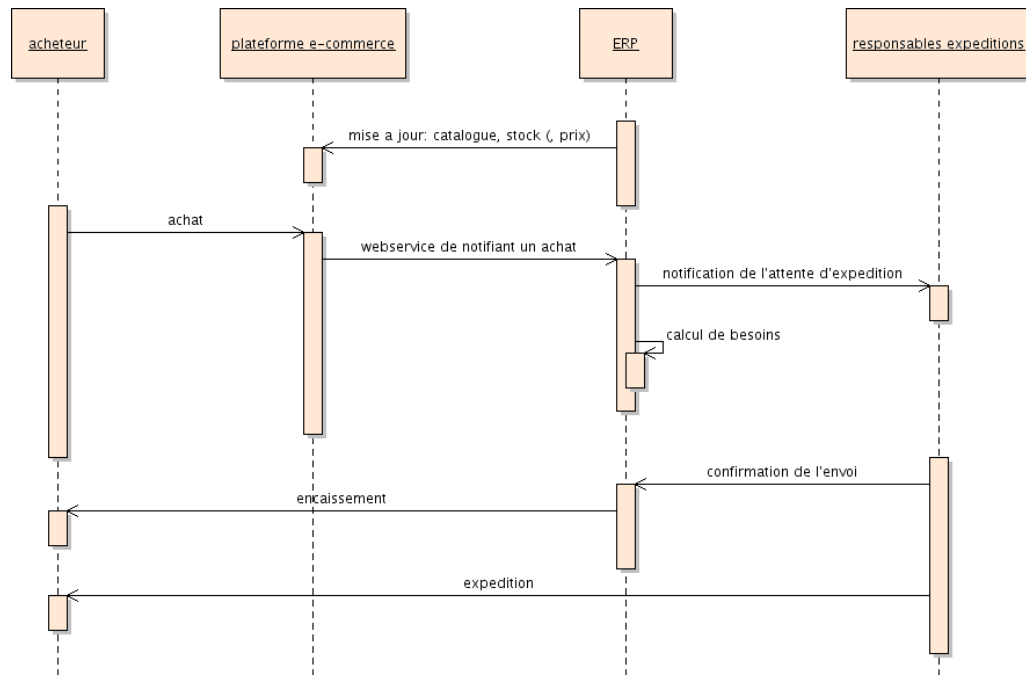
Au contraire, pour tous les autres ERP, il faudra s'interfacier avec un outil de vente en ligne spécialisé. Selon la popularité de l'ERP choisi, des connecteurs peuvent exister. En voici une liste non exhaustive susceptible de s'enrichir assez rapidement car ces connecteurs sont d'autant plus simples à développer qu'il suffit de s'inspirer de l'existant:

- TinyERP se connecte facilement à Joomla, OSCommerce et eZPublish; un autre connecteur Magento sera encore par ailleurs publié d'ici un mois par notre société Smile.
- Openbravo se connecte avec eZPublish.

Le fonctionnement standard est le suivant. Après avoir fait correspondre les catégories, moyens de paiement et autres paramétrages éventuels entre le frontal d'e-commerce et l'ERP, on utilise l'ERP pour gérer son catalogue de produits et de prix.

Régulièrement, on met à jour le catalogue du frontal e-commerce à partir du catalogue de l'ERP. Produits, fiches descriptives, prix et stocks sont donc importés. Par ailleurs, à chaque vente, le frontal d'e-commerce invoque un *webservice* notifiant la vente à l'ERP afin d'en tenir compte pour la comptabilité et la gestion des stocks.

ERP open source



9: COMMUNICATION ENTRE PLATEFORME E-COMMERCE ET ERP LORS D'UNE SÉQUENCE DE LA COMMANDE D'UN PRODUIT

Vente par points de vente (vente en caisse)



Un autre canal de distribution sont les points de ventes, par exemple des **caisses** de magasin. Dans ce cas, il faut pouvoir disposer d'un logiciel présentant le catalogue des produits et permettant la vente. Idéalement, ce logiciel est adapté aux **écrans tactiles**, a une très forte ergonomie, notamment en terme de gestion des cycles de focus et enfin est connecté à un **lecteur de code barres**. Un lecteur de code barre, permet

d'émuler l'entrée sur un clavier. Dès lors, la lecture d'un code barres s'apparente à l'entrée sur le clavier du code produit.

En termes d'architecture et de communication, le logiciel de vente en caisse s'apparente à la vente par Internet à ceci près qu'il peut y avoir plusieurs plateformes de ventes dans les différents magasins.

Notons qu'on peut également utiliser des puces **RFID** et un lecteur de puces en lieu et place d'un lecteur de code barres. Ce type

ERP open source

d'infrastructure sera surtout utilisé pour des mouvements de stocks en usine plutôt que pour de la vente, mais le principe d'architecture est assez semblable.

Dans TinyERP et Neogia, la gestion de vente par point de vente se fait par un module spécialisé qui produit les écrans adéquats. TinyERP semble plus abouti ici que Neogia car son client lourd en GTK³ sera plus approprié aux écrans tactiles et à la saisie de masse.

Mais c'est Openbravo qui va le plus loin ici avec une solution très mature: **Openbravo-POS**, anciennement **TinaPOS**, solution libre de vente en caisse réputée mondialement, désormais passée sous le girond d'Openbravo. Openbravo-POS va notamment beaucoup plus loin qu'un simple client lourd puisqu'il embarque beaucoup de logique propre de gestion des stocks ce qui permet de gérer une boutique avec plus d'autonomie vis à vis de l'ERP central. A noter: certains intégrateurs ont déjà connecté Openbravo-POS avec TinyERP.

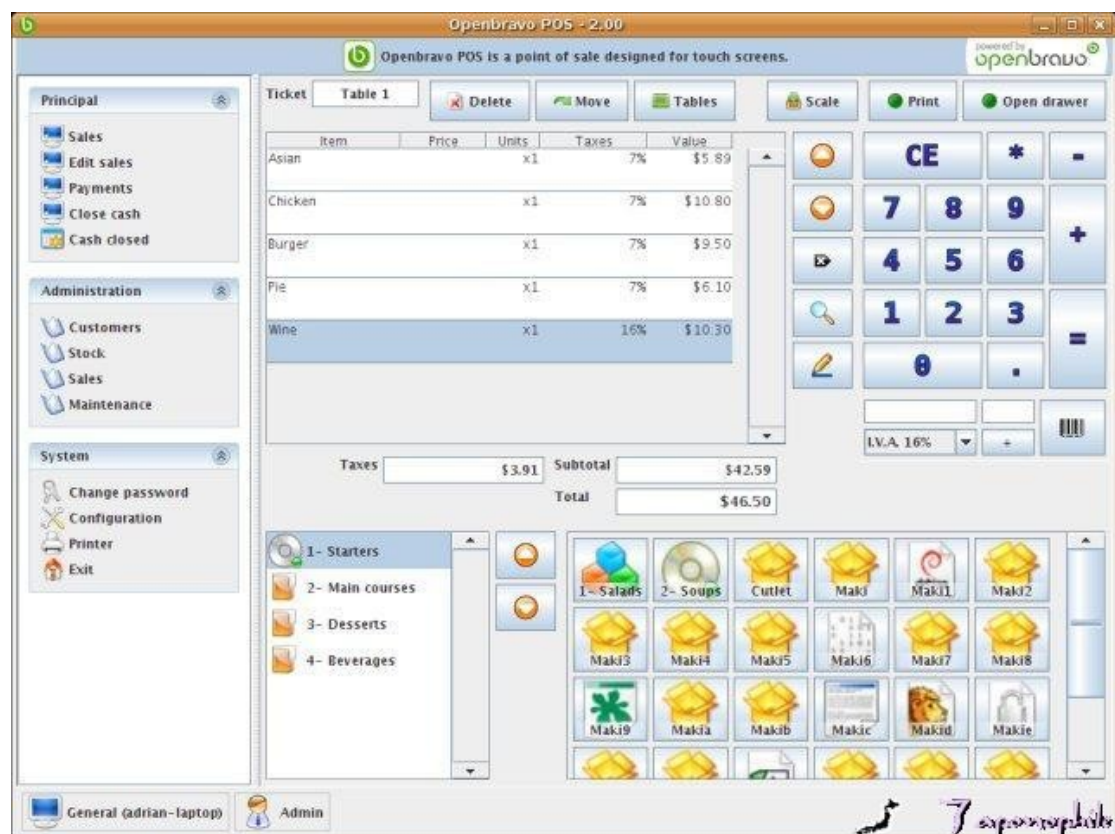


ILLUSTRATION 10: INTERFACE DE VENTE OPENBRAVOPOS

³²GTK (the Gimp ToolKit) est une bibliothèque libre réputée permettant de réaliser des interfaces graphiques ergonomiques sur toutes les plateformes.

Gestion des achats

De la même manière que notre entreprise vend des produits, elle peut en acheter à des **fournisseurs** et également **sous-traiter des services**. L'ERP permet de tracer ces transactions, d'en offrir l'intégration comptable et de mettre à disposition des **rapports** afin d'améliorer leur lisibilité et le pilotage des **budgets**. Le package de gestion des achats est souvent moins complexe que celui de la vente en général. Il permet principalement de suivre l'état de toute proposition d'achat et d'en déléguer correctement la validation au responsable achat ou au comptable.

Tout comme pour la gestion de prix de vente, c'est encore le **moteur de règles de prix** qui sera utilisé pour modéliser les prix d'achat en fonction des fournisseurs et des quantités achetées.

Les quantités à acheter peuvent justement faire l'objet d'**achats groupés** et on rentre alors dans l'ERP les quantités minimales à acheter par produit et par fournisseur.

11: PASSAGE D'UN ORDRE D'ACHAT SUR COMPIERE GPL

Il faut aussi noter que les ERP qui intègrent le calcul des besoins de production, c'est-à-dire la MRP1 (*Material Requirement Planning*), lancent automatiquement des propositions d'achats de pièces ou de sous-traitance en fonction du calcul des besoins. Selon l'ERP retenu, cette gestion de production est plus ou moins apte à planifier automatiquement les achats dans le temps, dans tous les cas il faudra valider le fonctionnement par des tests. Au minimum, les ERP retenus ici permettent de consigner manuellement les dates des achats à réaliser et

ERP open source

les délais d'approvisionnement dont il faut tenir compte. Ces délais seront notamment pris en compte lors de l'estimation des stocks futurs qui eux détermineront la capacité de production.

L'implémentation de règles pour choisir un item dans différents stocks ou s'approvisionner auprès d'un fournisseur particulier demandera des développements spécifiques dont l'ampleur dépend de l'ERP retenu.

Rappelons que les ERP abordés n'intègrent pas nativement de **configurateur de produits** à acheter et que seul un catalogue fonctionnant sur le même principe que pour les produits à vendre pourra s'appliquer aux produits fournisseur. On peut néanmoins faire l'achat d'un produit qui n'existe pas dans le référentiel et on se contente alors de le décrire par un commentaire. Au delà, intégrer un configurateur de produits dédié aux achats demandera des développements spécifiques conséquents.

De même, contrairement à des solutions spécialisées haut de gamme, les ERP open source n'embarquent pas nativement de solutions permettant la négociation sur des **marchés** puis la prise en compte de contrats à respecter (par exemple acheter telle quantité par mois à un fournisseur, quitte à anticiper des besoins qui ne sont pas encore ressentis par le calcul des besoins). Plus généralement, les ERP open source n'ont nativement pas de module d'**e-procurement** (catalogues électroniques des fournisseurs). Bien sûr, des ponts externes ou développements spécifiques sont là encore possible mais devront être pris en compte dans le budget ERP.

Gestion des stocks

Avec la gestion des stocks, on aborde déjà des concepts qui dépassent le périmètre des progiciels de gestion commerciale et comptable.

Les ERPs open source que nous avons retenus sont notamment **multi-entrepôts**, c'est-à-dire que les produits peuvent être stockés à différents endroits. Ces différents entrepôts ne sont pas nécessairement des stocks physiques distincts. En effet, dans une même entrepôt physique, on peut créer des 'entrepôts virtuels' dont le but est de répertorier, par exemple, les produits de l'entrepôt ayant subi un **contrôle qualité**, ou encore les **produits perdus** dans cet entrepôt.

Dans TinyERP et Neogia, les entrepôts peuvent être **organisés hiérarchiquement**: on voit donc les stocks consolidés d'un entrepôt globalement, mais on peut aussi 'zoomer' sur les sous-stocks qui le composent. Openbravo ou Compiere ont eux une notion d'entrepôt qui peut aller jusqu'à deux niveau de profondeur.

ERP open source

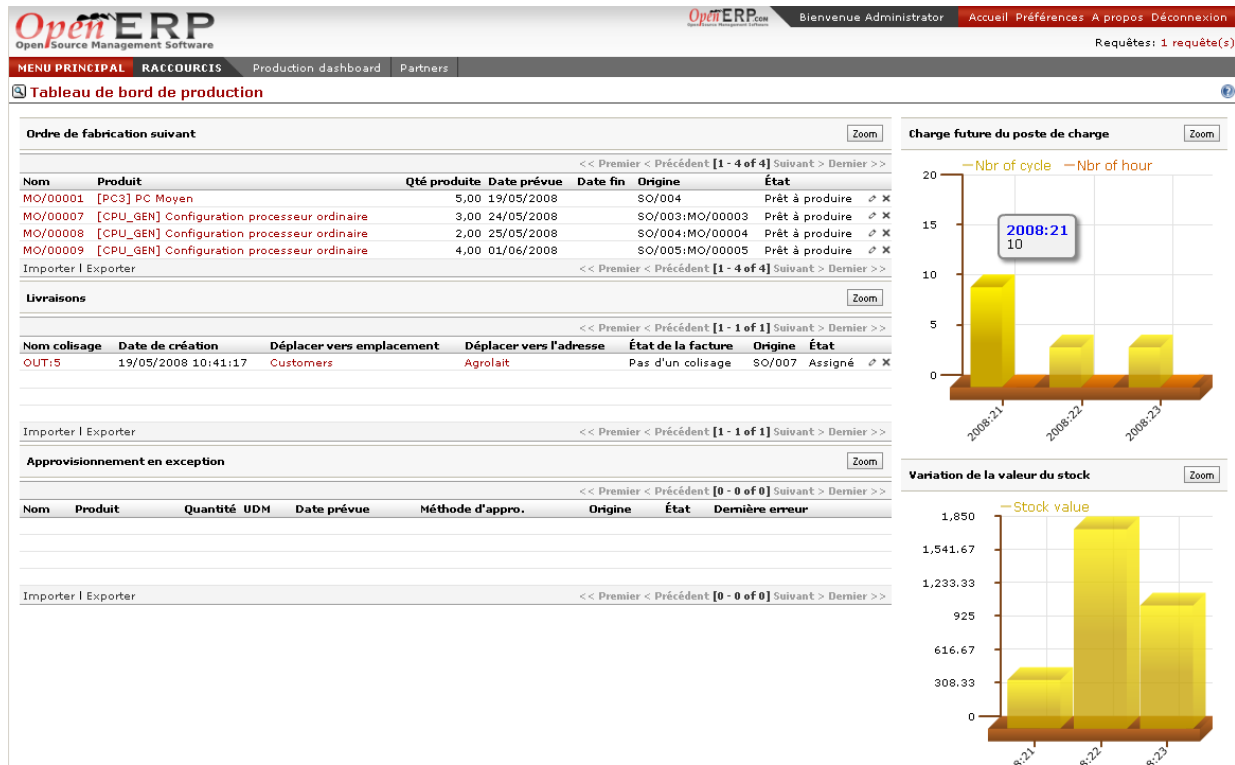


ILLUSTRATION 12: PRÉVISION DE LA CHARGE SUR LES DIFFÉRENTS POSTES DE PRODUCTION ET ÉVOLUTION DES STOCKS SUR OPENERP (TINYERP)

Nativement, avec la gestion des stocks, les ERPs étudiés permettent de consigner dans un 'stock virtuel' les **expéditions** pour les ventes et les **réceptions** des achats. La validation des envois et la réception des marchandises peuvent ainsi facilement faire l'objet de contrôles qualité ou être déléguées à des responsables particuliers.

C'est au moment de la validation d'un envoi ou d'une de réception de marchandise qu'est effectuée **l'assignation d'un produit à un entrepôt**. C'est-à-dire respectivement un prélèvement du stock pour une expédition ou un approvisionnement de stock pour une réception. Lors de l'assignation, l'entrepôt concerné est proposé automatiquement: le premier entrepôt ayant la quantité demandée pour une expédition ou le premier entrepôt à ré-approvisionner pour une réception. Cependant, la possibilité est laissée de forcer manuellement cet entrepôt. De même on peut choisir de **ventiler l'envoi ou la réception sur différents entrepôts**.

Notons encore une fois que l'on peut, **par des développements raisonnables, implémenter d'autres politiques d'assignation**. Par exemple privilégier l'expédition depuis un stock associé à un fournisseur moins cher.

ERP open source

Il est aussi important de noter qu'on peut avoir un produit encore dans un stock physique alors même qu'un client l'a peut être déjà commandé. Il faudra bien sûr en tenir compte pour ne pas le vendre à nouveau. Dans TinyERP par exemple, c'est le rôle des 'stocks virtuels' que de définir la disponibilité réelle d'un article.

Une autre fonctionnalité importante généralement associée à la gestion des stocks est la **traçabilité**. Dans TinyERP, Openbravo et Neogia au moins, on peut ainsi facilement afficher la traçabilité **amont** et **aval** d'un produit grâce à un numéro de série, c'est-à-dire qu'on peut réciproquement connaître la provenance exacte des pièces qui constituent un produit ou un lot, et pour chaque lot de pièces à l'entrée, on peut au contraire savoir par où elles ont transité et à quels clients elles ont été vendues.

Par ailleurs, les mouvements de sortie de stock peuvent suivre différentes méthodes de gestion automatique (ou encore d'allotissement) (**FIFO**, **LIFO**⁴, **index**, ...). Ces méthodes sont natives dans Neogia par exemple. En revanche elles ne sont encore qu'en phase de développement dans TinyERP et Openbravo. Attention, nous parlons ici des méthodes de sorties de stocks et pas des règles comptables de valorisation des stocks pour lesquelles tant TinyERP que la famille des Compiere offrent des solutions de LIFO/FIFO.

Gestion de la production

MRP1

La gestion de production est sans conteste la fonctionnalité la plus avancée d'un ERP. Sans prétendre faire un cours sur les ERP, rappelons les grands principes du calcul des besoins de production, soit la MRP1 (*Material Requirement Planning*). Pour les entreprises qui doivent assembler ou fabriquer les produits qu'elles vendent, la gestion de la production passe par **trois concepts essentiels: les nomenclatures, les gammes et les postes de travail**.

Les nomenclatures: il s'agit de la liste des pièces qui composent chaque produit, regroupées par opérations d'assemblage ou de fabrication. On y trouve donc la cascade des opérations qui part des pièces élémentaires et matières premières achetées et qui aboutit à la création d'un produit étape par étape. Ainsi, la nomenclature d'un ordinateur comprendra le boîtier, la carte mère, l'alimentation... De même la carte mère peut elle même avoir une nomenclature propre dans laquelle on trouve par exemple un processeur, une barrette de mémoire... En revanche, l'entreprise s'affranchira de définir la nomenclature du processeur si elle l'achète.

Les gammes: elles représentent, pour chaque opération de fabrication matérialisée par une association de pièces de nomenclature, l'effort à

⁴FIFO, *first in, first out*: premier entré, premier sorti. LIFO, *last in, last out*: dernier entré, dernier sorti.

ERP open source

fournir dans l'opération. On compte ici d'une part toutes les dépenses dont on doit tenir la comptabilité, autres que les pièces listées par la nomenclature, et d'autre part tous les paramètres liés à la cadence de l'opération dont la limitation pourrait induire une limite de productivité. On y consigne donc par exemple les heures de main d'œuvre requises, le temps d'immobilisation d'une machine, la consommation d'électricité par cycle. Idéalement, en cas de relèvement des exigences pour la préservation de l'environnement, c'est aussi là qu'on devrait entrer la pollution (CO2) et le coût environnemental (Tonnes Équivalent Pétrole, empreinte écologique...) de l'opération.

Les postes de travail: les opérations de fabrications, les gammes, sont assignées à des postes de travail. Ceux-ci sont physiques ou pas. On leur affecte des capacités de charge, par une limite du nombre d'heures de main d'œuvre par jour ou encore en termes de consommation électrique. Le but étant ensuite de vérifier que la planification de la production n'engendre pas de dépassement de ces limites et d'envoyer des alertes au responsable de la production le cas échéant.

Notons que parfois, nous désirons prendre en compte les coûts de fabrication d'une pièce intermédiaire mais que cette pièce n'entre en jeu que dans la constitution d'un unique produit. Ainsi, on ne veut pas lancer de calcul des besoins spécialement pour cette pièce puisque le besoin est directement rattaché au besoin du produit. Dans ce cas, on déclare que cette nomenclature est 'virtuelle', c'est par exemple le cas sur TinyERP.

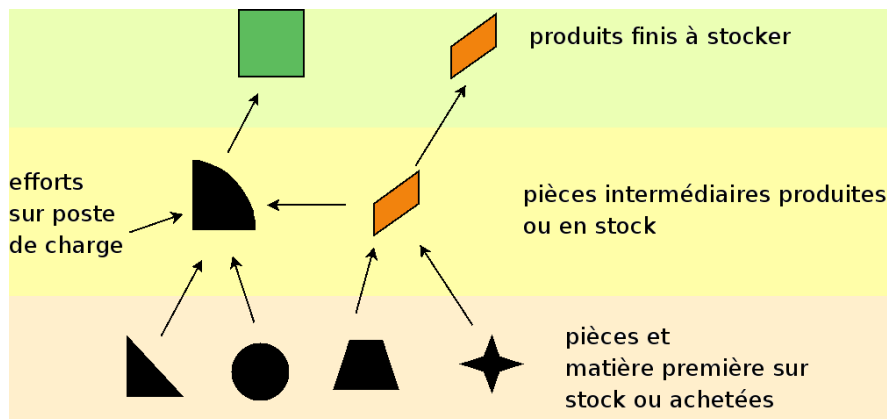


ILLUSTRATION 13: PRINCIPE DE LA DÉCOMPOSITION D'UN PRODUIT EN NOMENCLATURES

ERP open source

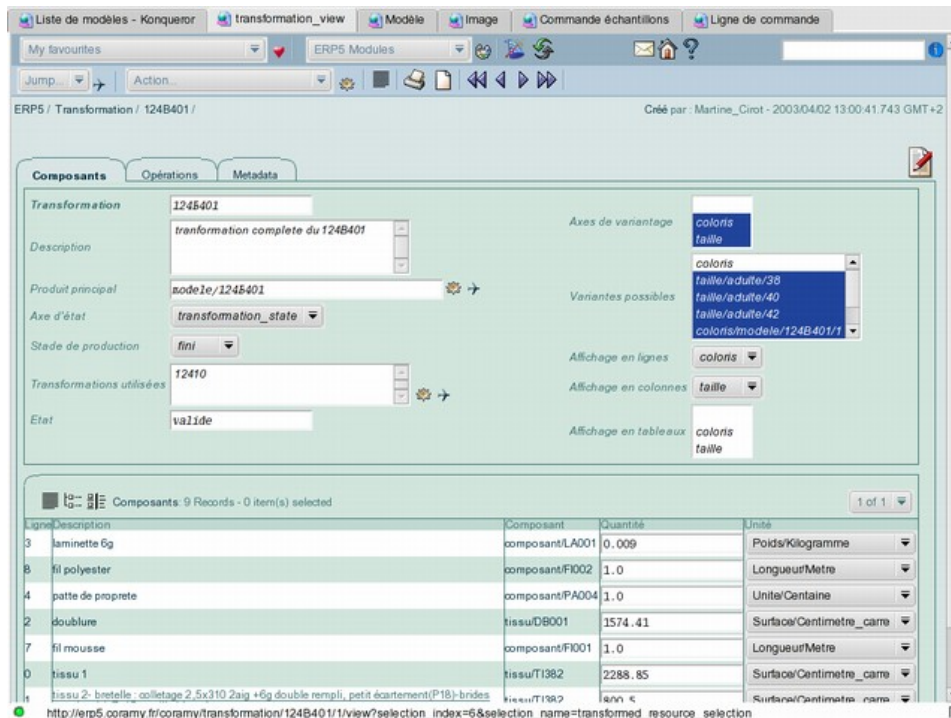


ILLUSTRATION 14: SAISIE D'UNE NOMENCLATURE DANS ERP5

En premier lieu, le paramétrage des nomenclatures et des gammes permet de calculer la **structure de prix d'un produit** en tenant compte de toutes les pièces incorporées et de toutes les opérations. Ceci sert par exemple à **établir la marge** par rapport au prix de vente. En effet, l'ERP est capable de calculer sur **l'historique des coûts** de production passés, le '*prix moyen pondéré*' d'un produit, c'est-à-dire le prix de revient moyen à des prix de revient variables selon les coûts à chaque instant, pondérés par les volumes de fabrication.

En second lieu, la gestion de production permet de lancer des calculs de **besoin en approvisionnement et en fabrication de pièces intermédiaires**. Pour ceci, le moteur de calcul des besoins tient compte à la fois des commandes clients, des prévisions de ventes, des stocks, et des prévisions d'approvisionnement et de leurs délais.

ERP open source

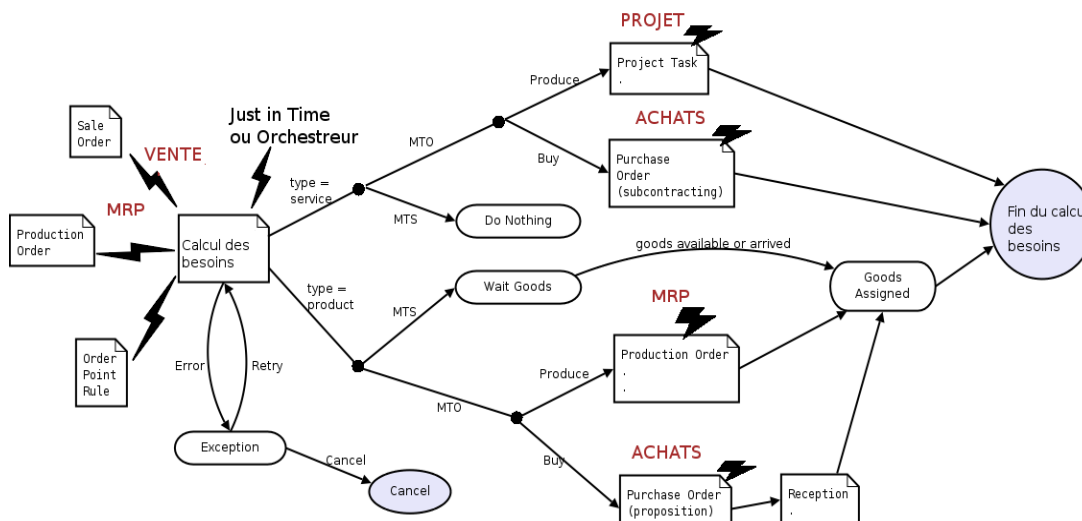


ILLUSTRATION 15: WORKFLOW DE CALCUL DES BESOINS (PRODUITS ET SERVICES) DANS TINYERP

Ce calcul des besoins peut être fait au fil de l'eau à chaque nouvelle commande (méthode du « **just in time** ») ou au contraire lancé **de façon périodique**. Le *just in time* permet de limiter les stocks au strict minimum mais peut poser des problèmes de puissance de calcul s'il y a trop de commandes et de produits ou pièces à gérer. Ainsi, une PME vendant de petits volumes à haute valeur ajoutée privilégiera le *just in time* alors qu'une grande surface ne pourra faire que des traitements périodiques, journaliers par exemple.

Ce dont nous avons parlé constitue donc ce qu'on appelle généralement la MRP1 que TinyERP, Openbravo et Neogia font très bien. ERP5 gère la MRP1 aussi mais nous n'avons pas eu le temps de le tester. Enfin Compiere a récemment développé la fonctionnalité de gammes et nomenclatures mais les retours d'expérience sont encore assez faibles et par précaution et en l'absence de transparence par une communauté active, il vaut mieux considérer que Compiere n'est pas encore au point sur la gestion de production.

MRP2

Au delà de la MRP1, on trouve la MRP2. La MRP2 a pour ambition, d'aller au delà du simple calcul des structures de coûts ou du calcul des besoins et de **planifier la production à l'optimum physique en tenant compte des ressources limitées des postes de charge**. Attention, la plupart des ERP libres ne rivalisent pas ici avec SAP R/3, Oracle Applications, Lawson ou encore certains ERP commerciaux très spécialisés.

Mais de très nombreuses TPE et PME n'ont pas une modélisation suffisamment fine de leur production pour que la MRP2 pratiquée dans les grandes industries leur soit utile. Dans les cas où la gestion de production fait beaucoup appel à des règles de gestion exceptionnelles, la

ERP open source

MRP2 n'est pas non plus bénéfique et tendra à apporter plus de contraintes que de solutions.

A défaut de savoir planifier la production Openbravo saura lui lancer des **alertes** sur les stocks qui atteignent leur minimum ou encore des postes de travail qui sont trop chargés. Openbravo ne sait pas planifier la production lui même, mais il offre néanmoins un référentiel pour consigner les futurs ordres de fabrications. Ainsi, il faudra planifier la production à la main, comme le font beaucoup de PME, ou avec un progiciel externe puis entrer ces ordres de fabrication dans l'ERP.

TinyERP gère aussi les alertes sur les stocks trop faibles mais va un peu au delà puisqu'il commence à s'attaquer à la planification de la production. Cette fonctionnalité émergente chez TinyERP pourra éventuellement être utilisée si elle convient après des tests. En effet, à ce jour, la planification de la production de TinyERP ne sait pas encore, par exemple, tenir compte des week-ends et des vacances.

On peut attendre aussi d'un ERP qu'il planifie la production à l'exact optimum des capacités physiques limitées de l'entreprise **en gérant des compromis entre les contraintes**, comme on peut le faire sur certains ERP commerciaux très spécialisés. On peut alors y définir certains **degrés de liberté** tels que la possibilité de compromis sur des délais dans le cas où certains postes de charge seraient en surcharge et qu'il faudrait **re-planifier la production..** Aucun ERP open source ne va jusque là pour l'instant. Il faut bien noter qu'on parle alors de problèmes complexes faisant appel aux mathématiques de la recherche opérationnelle, aux méthodes analytiques d'optimisation et à la programmation par contraintes. Des développements spécifiques en ce sens sont par ailleurs possibles mais seront assez conséquents.

Dans tous les cas, il est envisageable, d'importer dans l'ERP open source les ordres de fabrications depuis un planificateur externe qui sera un progiciel spécialisé ou un développement spécifique.

Simulations

Certaines entreprises seront également particulièrement intéressées pour simuler les besoins d'approvisionnement et de production liées à des commandes fictives. Ainsi, l'entreprise peut chercher à répondre à cette question: ai-je les capacités de produire cette commande? Vaut-il mieux privilégier tel ou tel client vis à vis de ma charge de production? Sur Openbravo par exemple, il y a une table où on peut entrer des prévisions de ventes faites par un outils externe (ou à la main) et qui donnent lieu à de tels calculs d'approvisionnements et de production simulés. Du côté de TinyERP, l'un des 200 modules est dédié à la prévisions des ventes selon des heuristiques simples. En revanche, la simulation des approvisionnements et ordres de fabrications qui en découleraient n'est pas native mais est en train d'être implémentée.

Gestion de projets

La gestion de projets est en certains aspects similaire à la gestion de production. Elle met néanmoins l'accent spécifiquement sur la gestion de **tâches** de durée variable et de **ressources humaines** essentiellement et n'a pas du tout besoin de faire appel aux notions de nomenclature par exemple. On s'attend en général à bénéficier d'interfaces de paramétrage et de monitoring spécialisées. Ceci explique qu'en général, la gestion de projet est un module un peu à part dans les ERP.

Parce que la montée en puissance du secteur tertiaire est plutôt récente, les ERP se traitent des problématiques de production industrielle avec plus de maturité que les services. C'est particulièrement vrai de la lignée des Compiere (Openbravo et Adempiere inclus), très inspirés de SAP. Pour cette famille, la lourdeur de la plateforme initiale a rendu plus difficile l'implémentation de modules efficaces de gestion de projet.

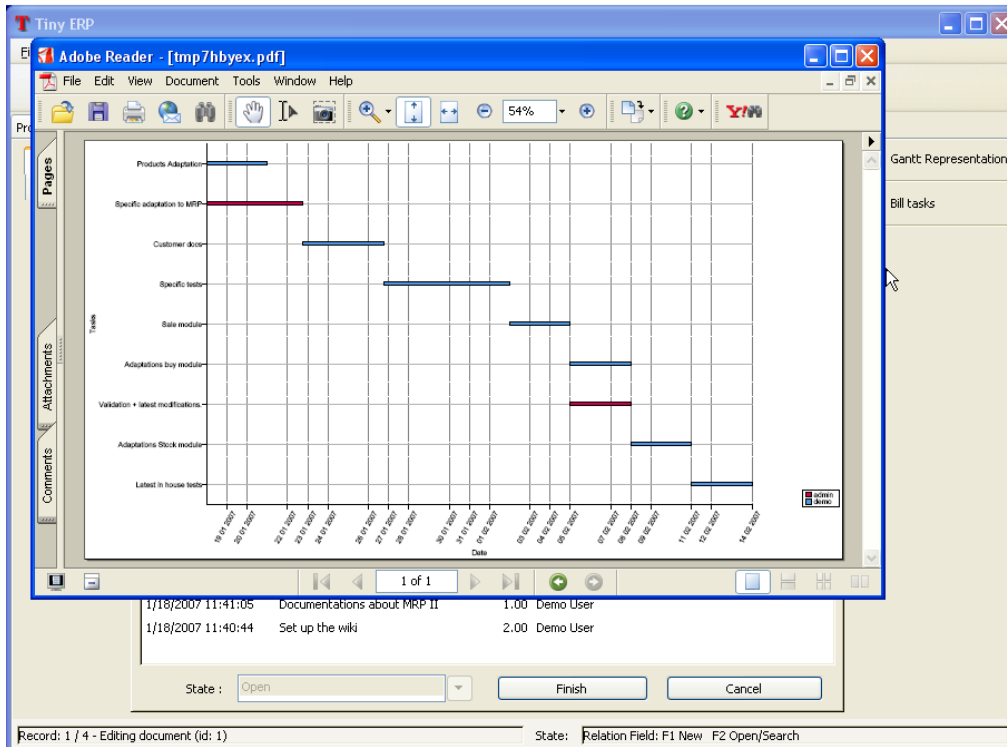
Dans Openbravo, par exemple, on peut consigner des projets faits de différentes tâches ou lots. Ces lots peuvent donner lieu à des lignes de facturation. Enfin un certain niveau de reporting permet d'**analyser la rentabilité du projet**, c'est-à-dire de comparer coûts réels imputés et budgets selon le degré d'avancement.

Cependant, par manque de module de gestion des ressources humaines, Openbravo ne permet pas à chaque employé d'être affecté aux tâches et d'imputer lui-même le temps qu'il passe sur chaque tâche. La tâche revient donc au chef de projet de collecter l'avancement du projet puis de le consigner dans l'ERP. On pourrait bien sûr passer outre moyennant des développements spécifiques. La gestion de projet n'y est donc implémentée qu'à un niveau macroscopique.

A l'inverse, dans Neogia ou TinyERP, par exemple, la gestion des RH, incluant les **feuilles de temps** et les **notes de frais**, permettent une assignation fine des tâches et projets.

Dans TinyERP, la notion de **tableaux de bord** personnalisés pour le monitoring par les **chefs de projets** (dont diagramme de **Gantt**) ou la **saisie des imputations** par les employés. TinyERP offre même une intégration des plannings avec Outlook ou Thunderbird.

ERP open source



ILLU

STRATION 16: DIAGRAMME DE GANTT DES PHASES D'UN PROJET SUR TINYERP

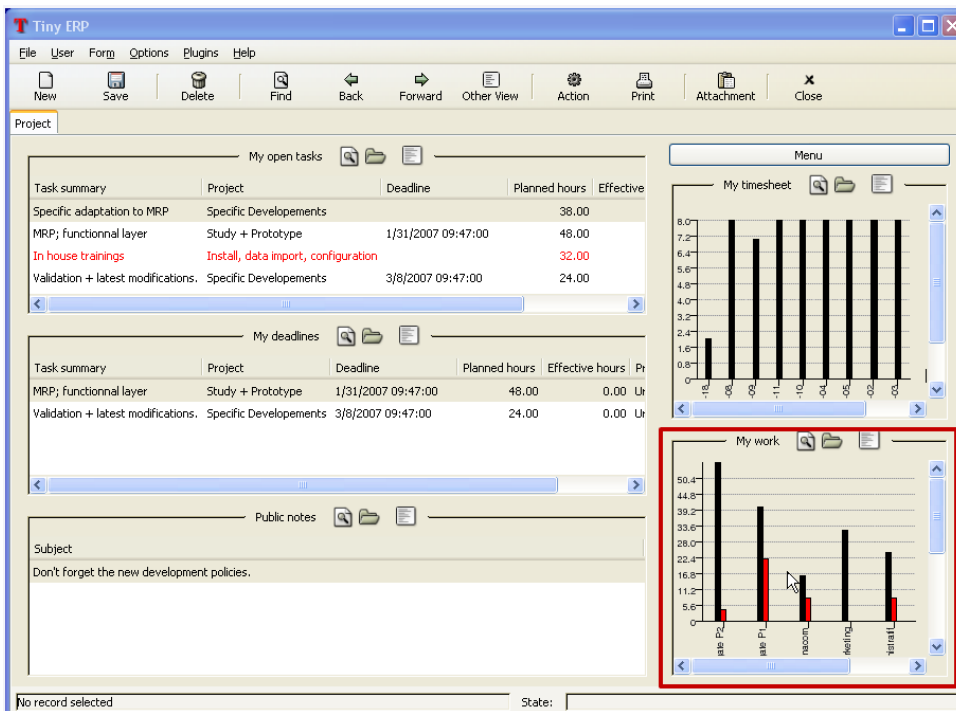


ILLUSTRATION 17: TABLEAU DE BORD DE PROJET PERSONNALISÉ DONT TÂCHES ASSIGNÉES, AVANCEMENT PAR TÂCHE ET PLANNING SUR TINYERP (ICI CLIENT LOURD, LE CLIENT WEB EST ISO-FONCTIONNEL MAIS OFFRE EN PLUS UNE VUE DU PLANNING CALENDRIER PLUS ABOUTIE).

ERP open source

La gestion de projet implique souvent une complexité organisationnelle mouvante en terme de **workflows d'entreprise**. Pour cela une modélisation de type **BPM** (*Business Process Management*) des process est idéale. TinyERP propose une modélisation BPM efficace des process qui garantit qu'ils seront simples à saisir, à maintenir, et surtout à respecter par les utilisateurs. Le concept utilisé par TinyERP est proche de la modélisation normalisée XPDL et permet l'import de fichiers de workflows XPDL. C'est à dire que la définitions des process d'entreprise est globalement portable mais pas autant par exemple que la norme BPEL, que malheureusement aucun ERP open source n'implémente à cette heure.

Openbravo vise quant à lui la modélisation BPM type BPEL mais il faudra attendre 2009 pour la voir en place nativement. En attendant, des intégrations au cas par cas avec des moteurs puissants et standards comme Intalio BPM sont envisageables dès lors qu'on aura pris soin de bien exposer les actions disponibles aux acteurs par web service SOAP.

Comptabilité

La comptabilité peut être décomposée en deux aspects: d'une part le respect de la législation avec la comptabilité générale et, d'autre part, le pilotage des budgets avec la comptabilité analytique.

Comptabilité générale

La comptabilité générale étant parfaitement normalisée et connue des professionnels, elle ne sera pas détaillée ici. Nous rappellerons juste que l'idée générale est de consolider, sur des périodes comptables données, soit des 'exercices comptables', les transactions financières effectuées sur les différentes comptes régis par les règles du **plan comptable général** et agrégés sur les documents légaux officiels tels que le **bilan** ou le **compte de résultats**.

Peu d'ERP libres sont matures en terme de compatibilité générale. En effet, même s'ils intègrent tous le plan comptable officiel, les ERP open source demandent de nombreuses mises au point quant au respect exact des regroupement de compte ou encore vis à vis du paramétrage correctes des différentes taxes, par exemple pour produire la **liasse fiscale**.

En fait, comme nous l'avons vu, les ERP libres ne peuvent pas être des projets à envergure nationale mais sont impérativement globaux, et à ce titre respectent plus ou moins bien la législation spécifique à un pays. Celle française n'étant souvent ni la plus facile, ni la première à être implémentée. De plus, la législation est changeante et elle suppose des mises à jour permanentes soit de la part d'une communauté active, soit d'un éditeur puissant.

ERP open source

Les sociétés cotées en bourse ont obligation d'utiliser un outil ERP certifié conforme à la norme comptable IFRS, mais ce n'est pas le cas des autres entreprises, et en particulier des PME, pour lesquelles la seule obligation est de respecter la loi, c'est-à-dire de produire une comptabilité valide tout en interdisant la modification directe par l'interface d'une écriture comptable clôturée. Ajoutons que si un logiciel commercial peut recevoir une certification (cas de Ciel ou Sage par exemple), cela n'est pas pertinent pour un logiciel libre. En effet, la valeur d'un logiciel libre résulte en premier lieu dans sa capacité de personnalisation et de mise à jour permanente. Or, une certification a un coût élevé, et n'est applicable qu'à une version donnée du produit.

Par ailleurs, si la plupart des entreprises n'ont pas d'obligation d'utiliser un logiciel certifié, dans beaucoup de grandes entreprises, le commissaire aux comptes refusera de prendre le risque, c'est-à-dire d'assumer lui même la responsabilité que la comptabilité soit erronée au lieu de se reposer sur la validation d'un organisme de labellisation.

Au final très peu d'ERP open source gèrent correctement la compatibilité française: c'est le cas presque exclusivement de Compiere, ERP5 et TinyERP, à condition d'y investir un temps important. Selon leurs éditeurs, Neogia et Adempiere produisent une comptabilité valide mais il est plus difficile d'en être certain à cause d'un nombre de déploiements en comptabilité française sensiblement moindre. Pour ce qui concerne Openbravo, même si le plan comptable français est inclus, il vaudra mieux attendre la seconde moitié de 2008 et passer d'abord par un pont comptable vers un logiciel spécialisé.

Heureusement les choses vont aller en s'améliorant et vont tendre à une comptabilité irréprochable. Pour ce qui est de TinyERP, il est même très probable que la comptabilité y devienne aussi simple que sur Sage courant 2009 étant donné le nombre important de personnes contribuant à l'édification d'une comptabilité plus immédiate, notamment de la part de l'éducation nationale à travers le réseau Certa qui commence à enseigner la comptabilité aux BTS avec TinyERP.

En l'absence de comptabilité générale française, il est aisé de bénéficier quand même des fonctionnalités de l'ERP, en mettant en place un **pont d'export des écritures comptables** vers une solution éprouvée et bon marché telle que Ciel, Sage ou Cegid.

Bien sûr, la mise en place d'un tel pont a un certain coût qui pourra être lourd pour une TPE. En général, un simple pont type fichiers plats CSV ou XML à importer par exemple tous les mois sera suffisant, surtout dans la mesure où on garde la comptabilité analytique sur l'ERP. C'est par exemple la solution en place à Smile pour une entreprise de 250 personnes et 14 M€ de CA.

ERP open source

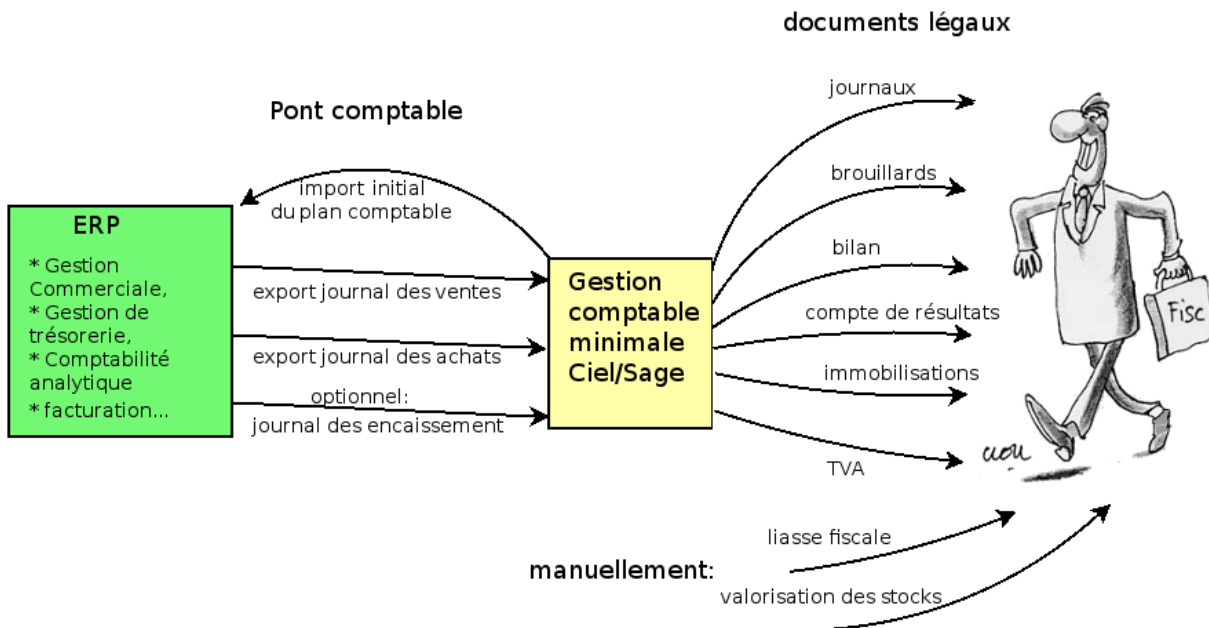


ILLUSTRATION 18: PRINCIPE D'UN PONT COMPTABLE. REMARQUE: AU LIEU D'UTILISER UN LOGICIEL DE COMPTABILITÉ, ON PEUT AUSSI TRANSMETTRE LES DONNÉES COMPTABLES ISSUES DE L'ERP À UN CABINET D'EXPERT COMPTABLE EXTERNALISÉ

Outre la production des documents officiels qui demande encore quelques contournements, comme nous l'avons vu, d'autres fonctions de compatibilité générale natives à l'ERP sont tout à fait utilisables. On y trouve donc: la gestion complète des paiements: brouillon, pro-forma, paiements avec différents types et échéances, paiements multiples, réconciliation des écritures, factures, notes de crédit, gestion de caisse.

De même, l'analyse financière est généralement utilisable, comme par exemple sur TinyERP ou Openbravo: analyse des créances et dettes, indicateurs de gestion, budgets sur les comptes généraux par exemple.

ERP open source

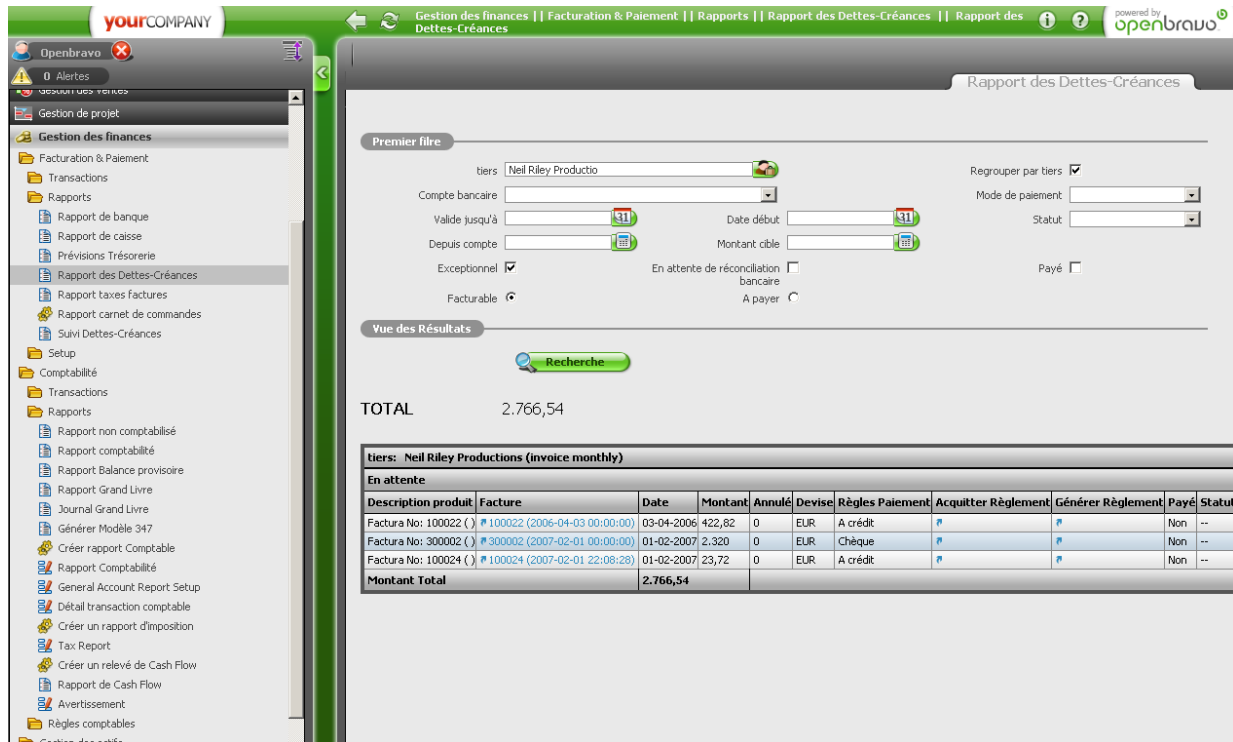


ILLUSTRATION 19: GESTION DE TRÉSORERIE SUR OPENBRAVO

Comptabilité analytique

Si la comptabilité générale reste encore un peu ingrate pour les ERP libres, la comptabilité analytique est au contraire un domaine où ils sont très compétitifs, à la hauteur des solutions de **reporting** réputées qu'ils utilisent, JasperReport ou Birt Report pour les ERP en Java notamment.

D'une manière générale, un produit ou une charge doit être associée à un ou plusieurs comptes analytiques de notre choix. C'est à dire que contrairement à la comptabilité générale, les critères d'analyses ne sont pas fixés par la loi mais laissent la place à la définition des axes les plus judicieux pour caractériser les performances de l'entreprise. Dès lors, on peut obtenir des rapports sur des périodes arbitraires et les comptes analytiques (les axes) de notre choix.

On peut aussi obtenir des comptes consolidés sur des groupements de comptes analytiques. En effet les comptes analytiques se comportent comme une **arborescence de catégories**. Ainsi, pour une entreprise ayant plusieurs entités, il est possible d'agrèger les coûts dans un premier niveau par entité géographique, puis, dans un second niveau hiérarchique, d'agrèger les coûts cette fois par centre de coût, à savoir par exemple l'administratif, la production, la vente...

ERP open source

Sur Neogia et TinyERP, la comptabilité analytique est à la fois multi-axiale et hiérarchique. Sur Compiere et Openbravo, la comptabilité analytique est multi-axe mais pas encore hiérarchique.

Dans TinyERP et Openbravo et Compiere au moins, on peut aussi créer des budgets sur des périodes arbitraires et regroupant des comptes analytiques de charges ou de produits arbitraires. On peut ensuite confronter les comptes analytiques à ces budgets au fil du temps.

Business Intelligence

Nous l'avons vu, aucun des ERP libres que nous avons étudiés n'embarque de solution native d'export de **datawarehouse** sous forme de **cube OLAP** qu'on pourrait associer ensuite à une solution de *business intelligence* telle que Pentaho ou SpagoBI. Bien sûr, certaines initiatives ponctuelles de constitution de cubes alimentés par l'ERP ont bien été déployées, notamment avec Neogia et Pentaho, mais il s'agit encore de développements spécifiques qui n'ont pas de généralité propre à les rendre facilement partageables pour tous les usages.

Cependant des plans d'actions sont actuellement mis en œuvre par les éditeurs pour répondre à ce besoin. Du côté de TinyERP, l'éditeur va publier à mi-2008 un plugin appelé '**OpenObject**' (basé sur le réputé SQLAlchemy) et qui permet d'introspecter les objets métier de TinyERP pour les exposer dans un cube OLAP. La solution de reporting associée permet donc de **changer les axes et d'opérer des coupes dans le cube de données** pour affiner les analyses. Ce plugin OpenObject, sera un framework open source qui pourra être utilisé avec ou au contraire indépendamment de TinyERP, un peu à la manière du module actuel de reporting de TinyERP: ReportLab.

Les avantages de cette solution sont:

- Il n'y a aucune configuration à effectuer: le modèle de données propre à l'entreprise est directement exposé dans le cube par introspection. Même si on adapte la structure de données au cas spécifique de l'entreprise, soit par paramétrage, soit par l'installation de plugins, la solution décisionnelle s'adapte automatiquement. C'est donc une source d'économie appréciable pour les TPE et petites PME.
- La solution décisionnelle tient compte automatiquement des politiques de droit définies dans l'ERP. Tous les utilisateurs peuvent donc bénéficier, à leur niveau de responsabilité, de solutions de reporting et d'analyse sans pour autant risquer d'accéder à des données confidentielles. C'est donc un moyen économique de distribuer au maximum la compétence, et donc de la maximiser, pour piloter votre entreprise.
- L'analyse est en temps réel sans avoir à se poser de question sur l'actualisation des données

ERP open source

Le fait que l'analyse soit en temps réel pourra néanmoins constituer au contraire un frein dans certaines grandes PME ou les grands groupes. Dans certains domaines, tels que la distribution, les flux de transaction de l'ERP sont si volumineux qu'effectuer des requêtes d'analyse décisionnelle en temps réel risquerait de mettre à mal la disponibilité de l'ERP pour gérer la production. Heureusement, divers niveaux de mise en cache de l'information feront reculer cette limitation du framework OpenObject.

De son côté, Openbravo annonce également pour la mi-2008, l'intégration d'un serveur de Business Intelligence qui sera soit **Pentaho**, soit **Jasper**; l'éditeur ne s'est pas encore déterminé. Cette solution sera plus standard qu'OpenObject et tiendra la charge plus confortablement, mais demandera en contrepartie plus d'efforts d'intégrations en terme de construction du datawarehouse ou de synchronisation des droits de consultation des données.

Quelle que soit la solution ERP retenue, il sera toujours possible de constituer spécifiquement un datawarehouse qu'on interrogera avec une solution de *Business Intelligence* standard type Pentaho ou SpagoBI mais ce développement spécifique sera plus coûteux qu'une solution intégrée bien sûr.

Un dernier point à considérer est de savoir si les requêtes de BI et de reporting sont effectuées sur les données de production en temps réel ou si on travaille sur des répliqués mises à jour régulièrement. En général, un système de réplication, s'il est suffisamment automatique et régulier, est suffisant pour donner des indicateurs assez précis pour un bon pilotage tout en laissant l'ERP gérer de gros volumes de transactions. C'est néanmoins un peu plus difficile à mettre en œuvre qu'une solution d'interrogation directe en temps réel des données de production type TinyERP avec OpenObject, peut-être plus adaptée aux petites entreprises.

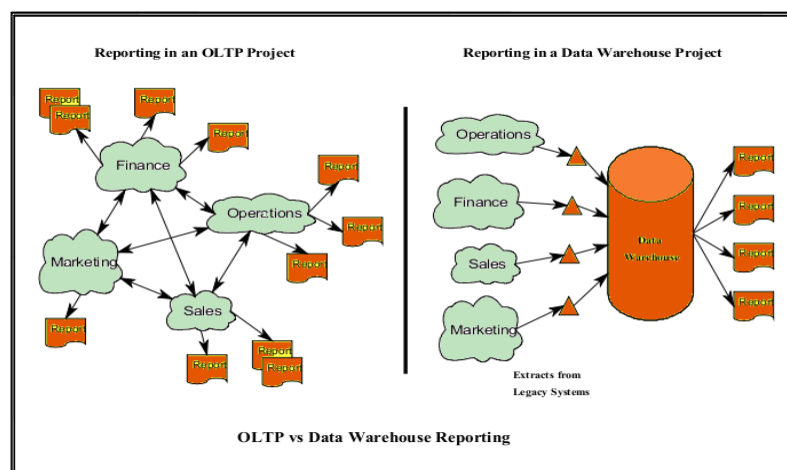


ILLUSTRATION 20: BUSINESS INTELLIGENCE: ARCHITECTURE TEMPS RÉEL À GAUCHE CONTRE DATAWAREHOUSE À DROITE

**CONCEPTS TRANSVERSAUX D'UNE
PLATEFORME ERP**

**Structure élémentaire de l'interface
d'un ERP**

On peut voir un ERP comme un **référentiel de données central**, sur lequel sont exécutés des traitements correspondant à des événements métiers. Dès lors, l'interface d'administration s'apparente en première approche à un explorateur de base de données qu'on aurait personnalisé.

Cette fonctionnalité d'explorateur de base de données est l'aspect le plus structurant de l'interface utilisateur d'un ERP. En effet si des produits propriétaires ou même certains libres très spécifiques se payent le luxe de développer chaque écran au bouton près, les ERP libres génériques adoptent eux un modèle générique de gestion d'entités, qui ne sera surchargé qu'en cas de besoin.

Et c'est ainsi que, tous les ERP retenus se structurent à peu près comme suit:

- Un menu de navigation qui permet de sélectionner un domaine fonctionnel, c'est-à-dire en fait un ensemble de tables et éventuellement une bibliothèque de traitements métiers associés.
- Un panneau permettant de visualiser au choix, soit la table (ou vue) sélectionnée, soit de montrer le détail d'une entité (une ligne) d'une table.
- Une barre d'outil permettant de faire les 4 opérations élémentaires sur une table: Création, Lecture, Modification, Suppression⁵.
- Une gestion du login et du compte utilisateur.

⁵ En anglais « CRUD »: Create, Read, Update, Delete.

ERP open source

yourCOMPANY

Openbravo Login

Sales Management || Transactions || Simple Sales Order || Header

powered by openbravo

toolbar de gestion des entites

Documt	Order Date	Business Partner	Price List	Currency
WO /06 / 73	11-08-2006	Chirika ,Company	Standard	EUR
WO /06 / 71	09-08-2006	Labaan ,Company	Special n°9	EUR
WO /06 / 36	28-04-2006	Simba ,Company	Standard	EUR
WO /06 / 11	13-12-2006	Yerodin ,Limited	Standard	EUR
RM /06 / 1	17-07-2006	Sine ,Corporation	Standard	EUR
Connecticut	02-01-2007	Chirika ,Company	Standard	EUR
Connecticut	06-11-2006	Fana ,Company	Fana Co.	EUR
Connecticut	10-08-2006	Chirika ,Company	Standard	EUR
Connecticut	04-08-2006	Chirika ,Company	Standard	EUR
Connecticut	28-07-2006	Chirika ,Company	Standard	EUR
Connecticut	27-06-2006	Yue You ,Corporation	Standard	EUR
Connecticut	25-04-2006	Siti ,Corporation	Standard	EUR
Connecticut	24-04-2006	Fana ,Company	Fana Co.	EUR
Connecticut	20-12-2006	Zawadi ,Limited	Standard	EUR
Connecticut	14-12-2006	Husani ,Corporation	Cullan ,Corp.	EUR
Connecticut	23-11-2006	Husani ,Corporation	Cullan ,Corp.	EUR
Connecticut	21-04-2006	Dabbous ,Corporation	Standard	EUR
CO - SOI071	19-12-2007	Actassi ,Corporation	Standard	EUR

Menu de choix de fonction

Vue tabulaire des entites

Pour chaque entité sélectionnée dans la table ou vue, on peut ouvrir un nouveau panneau qui permet de lire ou d'éditer l'entité en question:

yourCOMPANY

Openbravo

Sales Management || Transactions || Simple Sales Order || Header

powered by openbravo

toolbar de gestion de l'entite en cours

Document No. WO /06 / 7205 Order Date 11-08-2006

Business Partner Chirika ,Company

Price List Standard Currency EUR

Service Point Main Warehouse

Form of Payment Shipment Cash

Copy Product Template

Status

Summed Line Amount 6.97 Grand Total Amount 7.46

Document Status Under way Document Type Warehouse Order

Close

vue ou édition d'une entite en particulier

Le système de rôles

Afin de protéger l'accès à certaines données ou à certains traitements, un ERP utilise un système classique de *rôles*. Chaque utilisateur se voit attribuer un ou plusieurs *rôles*. Ensuite pour chaque table ou vue, on restreint les possibilités d'actions sur les entités (création, lecture, mise à jour, destruction) à certains rôles. Il en va de même pour le lancement de traitements. C'est l'union de l'ensemble des droits d'un utilisateur qui déterminent ses droits effectifs sur une entité donnée (modélisation dite **ACL**⁶). Attention, on distingue généralement la personnalisation des menus et fenêtres qui n'affecte que l'ergonomie, de la protection effective du modèle de données selon les rôles.

Sur Openbravo et sur la version à paraître cet été de TinyERP au moins, il est également possible de modifier l'interface de visualisation d'une même entité afin par exemple de masquer certains champs de l'entité à certains utilisateurs ou encore de masquer des boutons donnant lieu à des traitements qui ne sont pas autorisés pour l'utilisateur en cours.

Multi-sociétés

Openbravo, Neogia ou TinyERP sont multi-sociétés dans la mesure où les utilisateurs peuvent appartenir à différentes entités organisationnelles de l'entreprise. Dès lors, un certain nombre de paramétrages, comme la devise de travail, la langue de l'interface ou encore le type de plan comptable, s'appliquent selon la structure à laquelle l'utilisateur est rattaché. Sur Openbravo le caractère mutlisociété est natif et très bien intégré. En revanche, sur TinyERP, il s'agit d'un module d'extension dont la qualité d'intégration devra éventuellement être validée avant un projet d'envergure.

Ce fonctionnement multi-sociétés permet notamment le partage des référentiels et l'agrégation des données comptables sur l'ensemble des sociétés. Toutefois, en cas de grosses structures avec de gros volumes transactionnels, il vaut mieux parfois isoler l'ERP de chaque société, et importer, voire synchroniser un certain nombre de paramétrages de base comme les partenaires commerciaux ou le personnel, puis isoler le fonctionnement de chaque société, quitte à mettre en place des échanges de données entre sociétés par webservices.

Ce serait ce qu'on préconiserait par exemple pour une multinationale.

⁶ACL: *Access Control List*, liste de contrôles d'accès.

ERP open source

ILLUSTRATION 21: CHOIX DE L'ENTITÉ DE L'ENTREPRISE ET DE LA LANGUE LORS DE LA CONNEXION À L'ERP OPENBRAVO

Client web ou client lourd?

La base de données d'un ERP étant centralisée, quel est le mode d'interaction entre les utilisateurs distants et ce référentiel central ?

Historiquement, les traitements étaient des procédures stockées mises à disposition par le moteur de base de données et qu'on pouvait déclencher à distance. Cependant, le déclenchement de ces procédures et la lecture de ces données n'était pas fait par une interface web mais plutôt par des technologies dites "client lourd", c'est-à-dire une application spécifique, installée sur le poste de travail. Par exemple une application Java Swing pour Compiere, Microsoft MFC pour Sage ou encore GTK pour TinyERP.

Il faut dire qu'autour des années 2000, l'usage du web n'était pas encore aussi généralisé et que sans l'avènement des interfaces d'Ajax⁷, les interfaces web manquaient alors cruellement d'ergonomie. Impossible par exemple de bénéficier de l'auto-complétion de la référence d'un produit telle qu'on peut aujourd'hui la voir sur TinyERP ou Openbravo). Enfin, le web étant encore confidentiel, il était difficile pour un éditeur, à fortiori open source d'être compétent dans l'ensemble des domaines à la fois métier, technique et web.

Aujourd'hui c'est l'inverse qui se produit: **le web est incontournable**, les gens veulent pouvoir accéder à leur ERP de n'importe quel poste de travail, téléphones et PDA y compris, sans rien installer. Les utilisateurs

⁷Ajax: technologie de requêtes asynchrones qui mettent à jour la page web courante sans la recharger complètement, procurant une ergonomie améliorée.

ERP open source

sont aussi plus habitués à l'ergonomie du web (fonction recherche, bouton retour, marques-pages) qu'à celle des logiciels lourds et l'ergonomie ou encore l'esthétique de l'interface est plus aboutie en mode web grâce à l'émergence de standards et d'abstractions de haut niveau pour les manipuler (CSS par exemple).

Les ERP libres n'ont pas échappé à cette tendance. Certains ont même fait office de précurseurs. C'est le cas d'Ofbiz qui dès 2001 proposa une interface totalement web. Viennent ensuite Openbravo et TinyERP, Compiere étant à la traîne sur ce sujet, sauf dans son offre 'Compiere Professionnal' qui fait appel à un serveur web en technologie GWT mais qui n'est ni open source ni bon marché (500 euros/mois pour une base de 10 utilisateurs minimum). Néanmoins aujourd'hui aucun ERP ne peut venir sans une bonne interface web.

Si tous les ERP se mettent au web, tous n'offrent pas une interface web aussi aboutie que leur interface historique. C'est encore largement vrai pour Compiere dont l'interface web commence à peine à égaler l'interface lourde Swing qui embarquait sans doute trop de logique spécifique pour être facilement portée au web.

Quant à TinyERP, son module web est désormais iso-fonctionnel et éprouvé, ceci d'autant plus que le client lourd GTK d'origine n'embarquait aucune logique propre et communiquait déjà avec le serveur à la manière d'un navigateur web, mais par webservices XML/RPC. En conséquence, eTiny, le module web de TinyERP n'est qu'une surcouche très légère et très propre qui traduit ces messages en HTML.

OfBiz (et donc Neogia) et Openbravo n'ont eux juré que par le web depuis le début et la question d'un support partiel ne se pose pas chez eux. Concernant Adempiere, bien que la volonté de certains intégrateurs soit là, étant donné l'effort à accomplir et la gouvernance actuelle, il nous paraît très incertain de tabler sur la mise au point d'une interface web éprouvée aussi ergonomique et aussi fiable.

Il faut souligner que l'interface web ne présente pas de risque particulier en matière de sécurité puisque reposant sur des sessions sécurisées, par Http-S.

Pour être complet sur l'interface web, il faudrait encore évoquer les fonctionnalités difficiles à implémenter en mode web. Indéniablement, la BPM en est un. En effet, si on construit des workflows complexes, il faut bien les visualiser pour les monitorer correctement. Or la visualisation de graphe en mode web est un domaine encore jeune. Dès lors, soit on passe par des images du workflows non interactives, soit l'interactivité est au prix d'un client lourd type applet ou application java (comme sur Bonita BPM).

ERP open source

Reporting

Le reporting des ERP consiste à embarquer un moteur de rapports, un dictionnaire de rapports pré-programmés et de rapports personnalisés pour convenir aux usages standards. Comme par exemple imprimer un devis, une facture ou le carnet de commande.

Les ERP en Java abordés ici embarquent la solution standard **JasperReport**. Un rapport Jasper consiste en un fichier XML qui décrit de manière déclarative les éléments d'un rapport. Les parties dynamiques du rapport sont générées avec des requêtes SQL appropriées. De plus, les développeurs ou consultants peuvent facilement créer des rapports en utilisant l'outil graphique IReport.

TinyERP opte lui pour une solution similaire, basée sur le moteur **ReportLab**, un standard en Python, qui utilise un paradigme déclaratif XML similaire pour la programmation des rapports. Mais au lieu de se contenter de requêtes SQL de bas niveau, TinyERP met aussi à disposition dans chaque rapport l'objet métier associé, ses méthodes propres et bien sûr ses objets associés. Enfin, là aussi de l'édition graphique est possible grâce au développement communautaire d'un plugin OpenOffice permettant de composer graphiquement ses rapports comme on écrirait un document Word et en insérant des balises actives de reporting.

Ergonomie

L'ergonomie tient essentiellement à

- la navigation entre les écrans (les liens entre documents comme client et factures associées sont-ils intuitifs?)
- l'usage de raccourcis claviers (Tab, CTRL+S ...)
- l'auto-complétion à la saisie, à base d'Ajax
- La verticalisation de l'interface par rapport au métier de l'entreprise
- la présence de panneaux de contrôle personnalisés

Indéniablement, ce sont Openbravo, Compiere/Adempiere et TinyERP qui possèdent l'ergonomie la plus aboutie avec notamment des solutions de raccourcis personnalisés et d'associations entre documents.

La palme revient néanmoins à Openbravo qui offre en plus un design général très attractif et très professionnel et a également implémenté dans sa version 2.40 tous les raccourcis claviers habituels (CTRL+S, Tab...) dans son interface web.

TinyERP n'est pas non plus à la traîne avec son interface inspirée de SugarCRM, application à succès s'il en est. D'autre part, son interface alternative native GTK peut parfois s'avérer redoutable en terme

ERP open source

d'ergonomie de saisie lorsque celle-ci est capitale. On est alors sur des règles de gestion du focus et des raccourcis claviers proches de ceux du système opérationnel et qui se révèlent plus efficaces que le client web à la saisie.

En revanche, Neogia pêche clairement dans ce domaine avec une interface un peu « brut de fonderie », un peu trop uniquement axée sur les données et trop peu sur les utilisateurs.

TECHNOLOGIE ET ARCHITECTURES

Bien que ça ne soit pas le seul critère à prendre en compte, la qualité technique d'un ERP compte énormément. En effet:

- Elle détermine l'effort déployé pour coder les fonctionnalités existantes. Un outil au développement initial coûteux sera plus difficilement amorti.
- Elle détermine aussi l'effort à fournir pour étendre le périmètre: que ce soit réalisé en interne ou confié à un prestataire. Et ceci, qu'il s'agisse de développement ou de simple paramétrage.
- Elle détermine la facilité d'interfaçage de l'ERP avec les autres systèmes d'information. Par exemple avec TinyERP ou Neogia, les webservices sont natifs pour tous les objets métiers de l'ERP, fussent-ils personnalisés (architecture SOA). Dès lors, rien de plus facile que de l'interfacer avec d'autres systèmes.
- Elle impacte aussi la probabilité de bug et la facilité de leur résolution. En général, plus il y a de lignes de code, plus il y a de bugs. Néanmoins, si les abstractions ne sont pas des concepts répandus, alors il y a un certain risque à ne pas les dominer et à ne pas arriver à personnaliser ou corriger le code. Nous avons pressenti ce risque avec ERP5 et sa base objet Zope notamment.

Nous avons donc défini une série de critères dont l'expérience open source de Smile nous amène à juger qu'ils sont déterminants en matière de maîtrise des coûts de développement. Bien sûr, les critères que nous allons citer n'ont pas tous la même pondération et nous vous laisserons les apprécier à l'aune de vos besoins spécifiques. Par exemple, pour ERP5, en dépit d'un nombre de critères techniques assez positifs, nous avons vu des points très bloquants: sa base de données très exotique et sa politique de gouvernance qui n'engendre pas de vaste catalyse communautaire. Vous y remarquerez aussi qu'Openbravo se classe moins bien que Neogia. Pourtant nous aurions tendance à préconiser plutôt Openbravo pour un ERP clés en main pour des raisons d'ordre non technique, telles que fiabilité, finition, force de l'éditeur, force de la communauté d'utilisateurs.

Notez aussi que ces appréciations resteront valides pour quelques temps, car elles portent sur les fondements de ces plateformes ERP et non sur des fonctionnalités périphériques qui pourraient, elles, changer rapidement.

Détaillons alors à présent l'impact et la situation des ERP pour chaque critère.

ERP open source

Interfaces méta-générées à partir des données

Contrairement à beaucoup de solutions en PHP passées au crible, les interfaces de nos ERP ne sont pas « codées en dur », champ par champ, bouton par bouton.

Concevoir et coder de manière spécifique chaque page d'interface utilisateur pour chaque type de donnée représente un gaspillage de ressource qu'un logiciel libre ne peut pas se permettre.

Ils procèdent donc par introspection du modèle de données pour gérer par défaut les entités et au moins offrir les interactions standard de création, lecture, modification, suppression, sans code spécifique ou presque. Bien sûr, les ERP retenus offrent tous des dialectes XML pour personnaliser des vues ou bien créer des formulaires lorsque c'est nécessaire, mais du moins le principe n'est pas de coder chaque élément d'interface et son contrôleur associé. Ces avancées sont liées en particulier à l'héritage de Compiere et à son « *Application Dictionary* ».

Base de données relationnelle connue et robuste

Nous l'avons vu, un ERP a un fort besoin de garantir l'intégrité des transactions, notamment des flux financiers et d'inventaire. Afin de le garantir, il doit s'appuyer sur une base transactionnelle éprouvée, garantissant l'**intégrité transactionnelle** y compris en cas de plantage applicatif de l'ERP.

Notons à ce propos que peu de bases de données libres offraient de telles garanties au début des années 2000 lorsque beaucoup des ERP présentés ont été conçus. Par exemple, du côté de MySQL, l'intégrité transactionnelle n'est apparue qu'avec la version 5 en 2005. Voilà pourquoi peu d'ERP l'ont adoptée, à la différence par exemple des solutions de gestion de contenus. Le choix des ERP libres s'est donc généralement porté sur **PostgreSQL**, moteur offrant l'intégrité transactionnelle depuis l'origine, au même titre qu'**Oracle**. Néanmoins, pour des raisons historiques, Compiere avait lui choisi de cibler Oracle, à l'époque le seul SGBD toléré en environnement d'entreprise critique. Le portage de Compiere et ses dérivés à PostgreSQL date de 2006 et permet enfin d'envisager un déploiement open source robuste de bout en bout.

Tous nos ERP libres s'appuient désormais au moins sur une base relationnelle solide telle que PostgreSQL.

Seul ERP5 fait bande à part: il utilise une base dite objet, celle du framework Zope. Une **base objet** est un système dans lequel les données ne sont pas éclatées dans des tableaux plats reliés entre eux par des clés de correspondance comme dans une base relationnelle. Au contraire, un

ERP open source

peu à la manière d'un fichier XML, les objet métiers y sont sauvés avec une complexité arbitraire et la façon de requêter les données contenues dans ces objets est elle aussi très particulière

Pourquoi ERP5 a t-il fait ce choix? Parce qu'à l'époque où ERP5 a été conçu, il y avait deux choix simples: soit faire comme Compiere et renoncer à la modélisation métier objet pour s'offrir la robustesse et les performances relationnelles, soit opter pour la puissance de la modélisation objet et trouver un moyen de persister quand même des objets métier complexes et très éloignés des structures de donnée plates. ERP5 a donc adopté cette dernière solution qui a l'avantage de rendre très simple la persistance d'objet métiers arbitrairement complexes sans trop se poser la question ni des mapping relationnels, ni du moteur de persistance. Ce choix d'ERP5 est donc largement à l'origine du très vaste périmètre fonctionnel qu'on lui connaît.

Hélas, le moteur Zope n'est pas d'une aussi grande robustesse face à l'impératif d'intégrité transactionnelle. Et surtout, lorsqu'on fait des opérations relationnelles dans un ERP, comme par exemple en comptabilité analytique, sans même parler de requête OLAP, les performances d'une base objet s'effondrent littéralement. Pour y remédier, ERP5 a mis en place un système d'index relationnel (potentiellement géré par MySQL) en redondance de l'indexation de Zope. Mais au final, il en résulte une solution hétérogène dans laquelle il est difficile d'avoir une totale confiance et qui ne possède plus l'agilité séduisante de la base objet seule.

Nous pensons que la solution dans laquelle ont investi d'autres générations d'ERP récentes (TinyERP, OfBiz et Neogia y compris) a nettement plus de sens: il utilisent une **couche d'ORM**⁸ qui réalise le mapping entre la modélisation objet et la persistance relationnelle classique. Dès lors il tirent partie à la fois d'une modélisation objet puissante et d'une base robuste et connue telle que PostgreSQL. Notez que, cerise sur le gâteau, l'ORM de TinyERP est **scriptable**⁹ puisqu'il est écrit en Python. Dès lors, TinyERP permet un **cycle de développement très court**, idéal pour prototyper rapidement des solutions, sans passer par de longs cycles de compilation et déploiement. On peut ainsi charger une extension qui va étendre la modèle de données à chaud en production sans même avoir à arrêter l'application.

Ajoutons enfin que si les ORM de Neogia et de TinyERP ne sont malheureusement pas des standards reconnus tels qu'Hibernate, l'usage d'une base de données en clair et leur exposition native par webservice de tous les objets du modèle de données permet facilement de réaliser des modules d'extensions dans d'autres langages de programmation et qui ne font qu'envoyer ou collecter les bon messages auprès de l'ERP en le considérant un peu comme une boîte noire dont les subtilités n'auront pas nécessairement à être maîtrisées.

⁸ORM: *Object Relationnal Mapping*, mise en correspondance du modèle objet de l'application avec le modèle relationnel de la base.

⁹Scriptable: le code n'a pas besoin d'être compilé puis déployé, il peut être directement interprété à la volé, comme du code PHP ou PERL par exemple.

Tenue en charge et scalabilité

Lorsque le nombre d'utilisateurs simultanés ou que le nombre de transactions par seconde devient important, il faut anticiper la tenue en charge de l'ERP. Il ne s'agit donc pas d'un problème pour les TPE. En revanche, la plupart des grandes PME et grands comptes risquent d'y être confrontés.

Hélas, parce qu'avant de convaincre les grands comptes d'adopter des ERP libres, il faut d'abord que les PME les adoptent, les retours d'expériences sur de tels environnements à forte charge manquent cruellement. Nos considérations sont ici plus de l'ordre de l'extrapolation que de la certitude absolue.

Dans un premier temps, il faut voir quelle charge peut absorber une seule instance du serveur ERP et une seule instance de base de donnée, éventuellement sur une machine différente. Ce qui correspond à l'architecture classique et simple.

Sur de petites charges, il faut certes une plus grosse machine pour héberger un serveur Openbravo ou Compiere, notamment en terme de RAM (> 1 Go minimum) qu'un serveur TinyERP par exemple. En revanche, passé une certaine charge, Openbravo ou Compiere offrent de meilleures performances à machine égale.

Une analyse plus poussée montre que sur TinyERP, le grand nombre d'indirections du code dynamique en Python utilisé pour une modélisation de plus haut niveau d'abstraction se paie par une consommation de ressources significative au niveau du serveur applicatif et pas seulement au niveau de la base de données.

Heureusement, il est toujours possible de mettre en cluster très facilement ce serveur applicatif sur de nouvelles machines physiques. C'est à dire que les utilisateurs différents se connectent en fait à plusieurs serveurs applicatifs différents qui attaquent toujours la même base de données. Ceci est possible aussi bien pour la couche applicative des serveur Java que pour chacune des deux couches (web et applicative) dans le cas de TinyERP.

Néanmoins, on peut encore atteindre un tel niveau de charge que la mise en cluster du serveur applicatif ne suffit plus car le serveur de base de données devient à son tour le goulot d'étranglement principal. A ce stade, il n'y a pas d'autre choix que de mettre un serveur plus puissant ou de mettre le serveur de base de données lui même en cluster. Or, contrairement aux applications web type CMS, avec un ERP, on ne repousse pas très loin les limites car **la nature transactionnelles des traitements oblige ce cluster à travailler en mode multi-maîtres** et non pas maître-esclave. Dès lors, à partir d'un nombre limité de nœuds (4?), ajouter du hardware ne sert plus à rien car les nœuds consomment plus de ressources à synchroniser leurs informations qu'à traiter les requêtes des utilisateurs.

ERP open source

Dans ces situations critiques, il faut donc que l'ERP soit optimisé au niveau de ses appels SQL et de son modèle de données. Pour ce qui est de TinyERP, le recours presque systématique à l'abstraction objet avant d'effectuer un traitement et l'usage de PostgreSQL rend plus incertaine la performance native de cet l'ERP dans des environnements très exigeants. Ponctuellement il sera néanmoins possible d'optimiser certains points chauds identifiés en recourant à des procédures stockées, mais cela ne marchera que si ces points chauds sont isolés. Au final, nous pensons que TinyERP devra encore faire ses preuves pour être déployé dans de grosses PME ou dans des grands comptes.

En revanche, du côté des ERP descendant de Compiere (Openbravo y compris), la maturité et l'usage presque systématique des procédures stockées apportera des performances élevées. De plus, dans les cas critiques, ces ERP peuvent éventuellement être déployés sur une base Oracle, sensiblement plus performante que la base PostgreSQL. Ceci rend au final leur emploi particulièrement serein pour des gros déploiements dans les secteurs de la distribution, de la logistique ou de l'industrie.

Pour ce qui concerne ERP5, bien que l'indexation par un moteur MYSQL externe soit rapide, les considérations relatives à TinyERP sur la lenteur relative de Python (devant Java) et le tribut payé par la modélisation objet s'appliquent également. Les choses sont encore plus incertaines quant à la possibilité de maîtriser facilement la réplication de la base de données objet Zope.

Dictionnaire de structure de données dynamique

Pratiquement toute entreprise a ses spécificités fonctionnelles et donc ses propres structures de données. Par exemple, telle ou telle entreprise va vouloir rattacher un nouveau champ d'information (un « attribut ») à ses clients ou ses produits. Avec quelle facilité fait-on cela?

Tout le mérite de Jorg Janke, le créateur de Compiere a été de mettre en place un mécanisme, « l'**application dictionary** » permettant de faire ceci à la volée, sans avoir à rien coder. Ceci est rendu possible grâce à l'intégration de langage **DDL** (Definition Data Language) et d'une interface de personnalisation des structures de données. Dès lors on peut graphiquement et sans arrêter l'ERP, altérer la définition des quelques centaines de tables qui composent l'ERP pour l'adapter. De même une autre partie de l'interface graphique permet, elle, de personnaliser les interfaces de saisie pour s'adapter à ces nouvelles structures de données et ceci sans aucun code de programmation tant qu'on en reste à des champs de saisie et boutons permettant de manipuler les données simplement (créer, lire, mettre à jour, supprimer).

Aujourd'hui, tous nos ERP sélectionnés le permettent à l'exception de la lignée OfBiz (Neogia incluse) qui nécessitera, elle, d'éditer des fichiers

ERP open source

XML décrivant la structure des données (voire UML sur Neogia) puis de recompiler et de redéployer l'application. On se rend bien compte que s'il faut de la souplesse par rapport aux évolutions fonctionnelles ou bien même ne serait ce que pour minimiser les coûts de la personnalisation initiale, il vaut mieux que les choses puissent se faire de manière souple et dynamique. Dans tous les cas, les ERP étudiés embarquent des outils servant à sauvegarder et à restaurer les altérations du modèle de donnée, qu'on l'ait fait graphiquement ou non.

En revanche, toute nouvelle règle de gestion associée à ces nouvelles structures de données nécessitera bien sûr une étape de programmation. Pour les ERP en Java actuels, cela veut dire, passer par une étape de génération de code, compiler, puis déployer, ce qui est assez fastidieux. TinyERP de son côté raccourcit ces cycles puisque l'on pourra charger à chaud le module développé pour tester instantanément les modifications sur une réplique de l'environnement de production. En ce sens, ses capacités de paramétrages dynamiques dans des délais interactifs sont largement étendus.

Code métier basé sur le paradigme objet

Au delà de la souplesse dans la définition des structures de données, comme nous l'avons vu, la génération des Compiere (Openbravo et Adempiere compris), parce qu'elle est plus ancienne, a dû sacrifier la modélisation objet du code métier au profit de la performance de la persistance. A l'inverse les autres ERP retenus accèdent à la puissance du code objet à des degrés divers:

- un peu pour OfBiz: modélisation objet des structures de données, gestion objet du modèle relationnel
- un peu plus pour Neogia: possibilité d'héritage, de méthodes métiers rattachées aux objet, d'encapsulation
- parfaite chez ERP5 et TinyERP: toute structure de donnée persistante est un objet métier extensible, dont les responsabilités sont encapsulées et facilement identifiables.

Si le code fonctionnel n'est pas objet, il est procédural. Ainsi, dans la famille des Compiere, le code métier est assuré par un grand volume de lignes de procédures stockées en PL/SQL. Notons pourtant que des ERP commerciaux réputés tels que SAP R/3 ne bénéficient pas d'une technique de modélisation meilleure. Mais ceci a quand même quelques conséquences. En effet, si la modélisation objet a été inventée, c'est parce que le code procédural gérait mal l'**explosion de la complexité** (or un ERP est complexe) tandis que l'objet permet d'une part d'**encapsuler les concepts**, faciliter leur compréhension et **limiter leurs effets de bord** et d'autre part, de construire les structures de données et comportements métiers par héritage de cas plus génériques, ce qui factorise le code en

ERP open source

minimisant les risques d'erreurs et en **limitant les efforts à fournir**. Dès lors, ces ERP au code procédural se prêtent moins bien aux extensions fonctionnelles.

Gageons qu'Openbravo, avec sa forte dynamique de développement, saura attirer les talents et les efforts communautaires pour dépasser cette limitation. Openbravo a notamment recruté comme chef de produit, Paolo Juvara, ancien architecte de l'ERP Oracle Applications, avec lequel nous avons pu échanger et qui nous a fait part de son ambition pour la nouvelle plateforme d'Openbravo dénommée **Green**. La plateforme Green passera par l'ORM **Hibernate**, un standard du monde J2EE, ce qui donne une perspective très satisfaisante à moyen terme.

Il n'est en revanche pas sûr qu'Adempiere, projet très communautaire, ait la force d'affronter ces défis. Du côté de Compiere, le manque de dynamique de conquête et de transparence dans la gouvernance ne permet pas non plus d'envisager ces évolutions sereinement en l'état.

Webservices (SOA) fournis en standard par la plateforme

Il est capital que l'ERP puisse, en cas de besoin, être connecté avec d'autres progiciels plus spécialisés: logiciel comptable classique, CRM pointu, planificateur de production, outil de prévision des ventes, outil de prévision des budgets...

Or il existe essentiellement 3 façons d'interfacer un ERP:

1. Par imports de fichiers plats CSV ou XML. Ceci marche et est envisageable par exemple pour les ponts comptables mais cette solution ne sera jamais adaptée aux problématiques d'intégration en temps réel ni même aux architectures de notification/subscriptions à des événements.
2. Par accès SQL. En effet 2 applications différentes peuvent bien écrire et lire dans la même base de données et dialoguer ainsi. Même sur ERP5 qui n'utilise pas une base relationnelle classique, ceci est rendu possible grâce à une panoplie de connecteurs SQL fournie par l'éditeur. Cette méthode a néanmoins des limites. En effet, elle ne délègue pas à chaque progiciel la gestion des contrôles et des règles de sécurité des données. Seuls éventuellement les 'triggers' (déclencheurs) de la base de données peuvent jouer un petit rôle mais l'essentiel de la logique déléguée à la couche logiciel d'accès aux données reste court-circuitée. C'est donc faisable, mais au lieu de se reposer sur des API simples, il faudra à chaque fois se plonger dans les modèles de données très complexes pour être sûr de ne faire aucune erreur. Enfin, cela ouvre souvent des failles de sécurité sur des données pourtant critiques.

ERP open source

3. La meilleure méthode est donc l'échange par webservices. C'est à dire un peu à la manière dont votre navigateur web interroge le serveur ERP mais suivant cette fois des protocoles et des formats dédiés à la communication entre programmes. Il s'agit donc essentiellement des normes XML/RPC et SOAP. XML/RPC a l'avantage d'être très simple. SOAP permet lui une introspection automatique et normalisée (WSDL) des services par des logiciels externes mais ceci se fait au prix d'une plus forte complexité.

Dans la lignée Compiere (Openbravo et Adempiere), les webservices SOAP ne sont pas généralisés nativement par la plateforme. Ils sont présents sur des domaines tels que le catalogue des produits ou la vente d'un produit, mais pour les autres domaines ou pour des altérations du modèle de données, il faudra développer spécifiquement de nouveaux webservices selon les besoins, afin d'exposer un objet métier.

A l'inverse, la lignée OfBiz, ou encore TinyERP, sont nativement orientés **SOA**, c'est-à-dire, qu'à leur cœur, presque au niveau de l'ORM, tout objet stocké dans l'ERP est exposé en webservice, en respectant les règles d'habilitations définies sur cet objet. Ceux qui connaissent le monde des ERP propriétaires reconnaîtront ici l'équivalent du fameux module **Netweaver** vanté par SAP pour promouvoir les versions plus récentes de R/3. OfBiz ou TinyERP vous offrent la même chose, gratuitement.

S'interfacer devient alors enfantin et ce, depuis n'importe quel langage de programmation. Dans TinyERP, même la modélisation des workflow BPM est accessible par webservice très simplement. Dès lors il est aisé d'avoir une tierce application qui, par exemple, fasse avancer un workflow de vente ou de gestion de production définie dans l'interface de TinyERP. Ceci devrait gommer les réticences que l'on pourrait avoir vis à vis de TinyERP et du fait qu'il soit codé en Python, un très bon langage, abondamment utilisé chez Google par exemple, mais qui a le défaut d'être moins connu en France que Java ou PHP par exemple.

Modélisation BPM de haut niveau

Les processus d'entreprise peuvent être complexes et changeants. Ce serait par exemple le cas d'un processus d'attribution de réduction commerciale mettant en jeu différents niveaux de la hiérarchie commerciale intervenant pour tenter successivement d'arriver à un accord commercial à chaque étape de la négociation. Ainsi, chaque commercial peut avoir, à son niveau, le droit de proposer une réduction. Si ceci ne suffit pas à convaincre le client, alors le niveau hiérarchique supérieur peut prendre le relais en proposant une promotion supérieure s'il le juge stratégique.

Dès lors avec quelle lisibilité et quelle flexibilité ce type de processus est-il mis en œuvre ? Dans quelle mesure, pourra-t-on le changer demain pour répondre à un nouveau besoin ?

ERP open source

Il y a deux types de possibilités:

- Ou bien aucune abstraction des process n'est proposée et le process est éclaté en un ensemble de méthodes (les transitions du workflows) qui s'appellent entre elles ou doivent être déclenchées manuellement. L'état du process est alors matérialisé par un ensemble d'indicateurs dans la base de données et il revient au développeur de bien traiter les différents cas possibles, et d'éviter les situations de blocage et les erreurs. Dans un tel cas, on ne dispose pas d'une bonne visibilité quant à l'état d'un processus, ni d'une bonne facilité à changer un processus. De plus, à cause de cette difficulté, un ERP de cette catégorie propose nativement un ensemble de processus assez réduit. Dans ce cas, il est alors plutôt du devoir de l'utilisateur de bien exécuter, dans le bon ordre et selon la situation, les différentes actions disponibles.
- Au contraire, l'ERP peut faire appel à une modélisation générique de type BPM qui
 - rend déclarative la programmation de ces process, et donc en facilite considérablement le codage, la maintenance et les modifications,
 - offre une lisibilité globale de l'état des process et des différentes options autorisées à chaque étape,
 - décharge totalement l'utilisateur des contraintes techniques du respect des procédures et le laisse se concentrer sur la stratégie fonctionnelle.

Des ERP libres que nous avons passés en revue, TinyERP, ERP5 et dans une moindre mesure Compiere utilisent ce type de modélisation BPM. Dans Neogia/OfBiz, ce type d'abstraction existe aussi mais n'est accessible que par l'intermédiaire du code et n'autorise pas en l'état la souplesse nécessaire pour réellement apporter de la souplesse à l'entreprise.

Pour ce qui est de TinyERP, la puissance de la modélisation BPM est très satisfaisante, n'ayant rien à envier au module BPM de SAP R/3 par exemple. Un processus de vente, un calcul des besoins, la réconciliation d'une facture et de nombreux autres processus deviennent totalement **monitorables et flexibles**. Si monitorer l'état d'un processus est à la portée de n'importe quel utilisateur averti, modifier un processus est toutefois un peu plus ardu et s'adresse plutôt au programmeur ou au consultant fonctionnel. Mais le niveau de compétence exigé pour le faire reste très faible. Que la définition des processus soit réalisée en interne ou confiée à un prestataire, dans tous les cas la productivité se traduira en réduction de coûts.

Openbravo n'a ici pas repris le code de Compiere pour des raisons de licences incompatibles sur cette partie récente de Compiere (GPL pour Compiere, MPL pour Openbravo) et aussi parce que la solution de Compiere n'était pas non plus si séduisante techniquement parlant.

ERP open source

Openbravo vise donc le passage par un moteur BPEL externe, tel Intalio ou Netbeans pour définir et exécuter ses process dans sa nouvelle plateforme 'Green'. Mais ceci ne sera disponible que courant 2009.

Modularité de la plateforme

Nous l'avons vu, un ERP peut devenir extrêmement complexe et s'étoffer de nombreuses fonctionnalités liées aux besoins des différentes verticalisations métier.

Or tous les usages n'ont pas besoin de toutes les fonctionnalités, il faut donc absolument pouvoir s'affranchir facilement de leurs diverses interfaces et paramétrages.

De plus, un éditeur ne pourra pas apporter seul toutes les briques fonctionnelles nécessaires à chacun. D'autant plus dans le modèle open source où le but recherché est la juste mutualisation des efforts et non la captivité des tous les utilisateurs envers un éditeur unique.

Dès lors, il est capital de disposer d'une plateforme modulaire permettant, tant à vous qu'à de véritables **acteurs tiers** du marché, de développer et distribuer facilement des extensions fonctionnelles¹⁰.

Mais disposer d'une plateforme modulaire ne se décrète pas. On rencontre nombre de logiciels libres qui maîtrisent les aspects métiers mais négligent l'infrastructure de leur plateforme. La modularité nécessite de **fortes compétences en architecture logicielle**, qui sont rarement maîtrisées par des acteurs très proches de leur métier fonctionnel. Si cela a peu de conséquences sur les ERP commerciaux au développement naturellement monolithique, **dans l'open source un logiciel se doit d'être modulaire**, c'est une qualité vitale.

Pas d'exception avec les ERP: très peu d'entre eux ont des architectures dont la modularité actuelle est suffisante. En fait, encore une fois TinyERP fait la course en tête loin devant avec **plus de 200 modules d'extensions** dont une bonne moitié ont été développées par des tierces parties. En effet, grâce à son fort niveau d'abstraction, tant grâce à ses vrais objets métiers qui peuvent être étendus par simple héritage, que par sa modélisation BPM, il minimise la quantité de code fonctionnel requis et rend plus fine la granularité avec laquelle différentes fonctionnalités peuvent s'intégrer. Il rend ainsi très aisée et lisible l'écriture de modules fonctionnels, notamment en s'inspirant de l'existant. Mais surtout, chacun des modules déclare des dépendances à d'autre modules et TinyERP gère donc le graphe des dépendances, en évitant les cycles, à la manière de l'IDE Eclipse ou encore d'une bonne distribution Linux.

Mieux que ça, grâce à la scriptabilité de sa plateforme, TinyERP permet aussi de charger à chaud des modules fonctionnels sans rien recompiler

¹⁰ Voir notre livre blanc « Introduction à l'open source » pour mieux cerner ces mécanismes de symbiose entre les acteurs économiques du libre)

ERP open source

ni même arrêter la production. Idéal pour charger soi-même l'un des 200 modules après l'avoir dûment testé sur une réplique de la base de production. Enfin, TinyERP offre même une interface conviviale de gestion des modules qui peuvent être pris en libre service depuis les dépôts listés (et modifiables) selon la politique de version décidée . Ne vous engageons à ne pas nous croire sur parole mais à constater vous-mêmes la vitalité de ces développements tiers en consultant les forums de TinyERP.

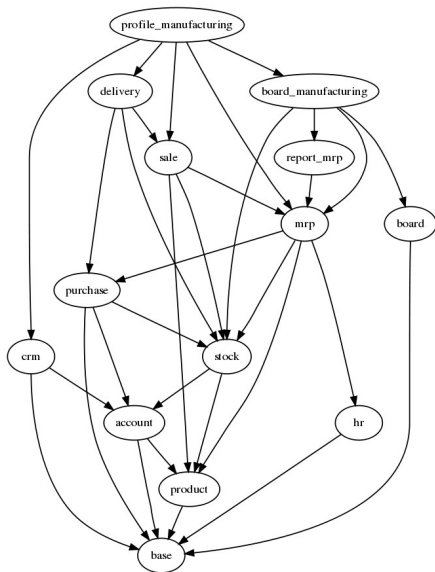


ILLUSTRATION 22:
MODULARITÉ SUR
TINYERP: LA GESTION
DE PRODUCTION DÉPEND,
ENTRE AUTRES, DE
MODULES TELS QUE LES
LIVRAISONS, LES VENTES,
LA MRP, LES ACHATS
ETC...

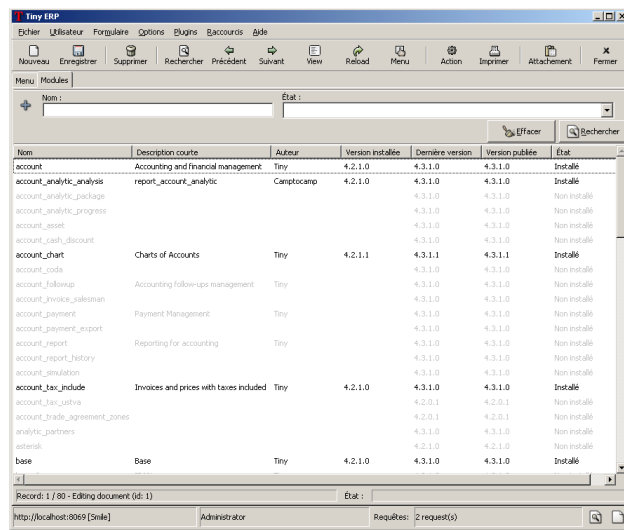


ILLUSTRATION 23: DE FAÇON PLUS OPÉRATIONNELLE, VOICI L'INTERFACE DE TINYERP, ICI DANS LE CLIENT LOURD, QUI DONNE À L'ADMINISTRATEUR LA POSSIBILITÉ D'INSTALLER UN MODULE OU UNE GRAPPE DE SELON LE RESPECT DE CES MÊMES DÉPENDANCES.

Pour ce qui est des autres ERP, la situation de la modularité de la plateforme n'est réellement pas au même degré de maturité, mais elle n'est pas catastrophique pour autant. Jusqu'à un certain point on peut installer des extensions tierces. Les différences sont alors que ces modules tiers:

- ont **une granularité assez forte**, surtout sur la lignée de Compiere, ceci car l'unité de code ré-utilisable, **la procédure stockée est beaucoup moins fine et rarement réutilisable à cause d'effets de bords difficiles à maîtriser**.
- sont difficilement chargeables à chaud. C'est à la rigueur possible dans la lignée Compiere tant qu'aucun code Java n'est requis (juste

ERP open source

des structures de donnée et des procédures stockées), mais pas possible sur la lignée OfBiz

- leur dépendances ne sont pas gérées, à vous d'installer ce qu'il faut et autant dire qu'il s'agit d'un travail d'expert.

En conséquence, l'écosystème des modules packagés et prêts à l'emploi est réduit sur ces ERP. Cela n'empêche pas que sur la lignée OfBiz un grand nombre de bouts de codes puissent être intégrés par des développeurs. Et encore une fois, Openbravo adopte ici une démarche pro-active en anticipant le passage aux bibliothèques réputées **Spring** ou **OSGI** pour mieux gérer le chargement d'extensions dans sa future plateforme Green à paraître courant 2009. Et d'ores et déjà, depuis la version 2.40 d'Openbravo, il est possible pour une tierce partie de packager des modules d'extension, même si la granularité de ceux-ci laisse encore à désirer et ne laisse pas encore augurer de la réutilisation d'extensions tierces dans de nouvelles extensions.

ERP open source

PREMIERS CRITÈRES DE SÉLECTION

Comme on l'a vu, un ERP est un système complexe, dont le modèle de données est composé de plusieurs centaines de tables. De plus, le choix de l'ERP est très structurant pour l'entreprise: d'une part les coûts de mise en œuvre sont importants et d'autre part l'ERP a vocation à héberger toutes les données critiques de l'entreprise. Cela implique un certain nombre de précautions quant à l'architecture technique choisie et quant à la santé de l'éditeur et ou de la communauté du logiciel.

Taille et dynamique critique

Nous avons d'emblée rejeté toute solution n'ayant pas suffisamment **fait ses preuves**. Nous avons conscience de l'injustice ainsi faite aux excellentes solutions techniques, notamment basées sur Hibernate ou Spring qui n'ont pas encore acquis leurs lettres de noblesses telles qu'OpenBlueLab, JFire ou PlazmaERP. Mais on ne peut prendre des risques sur un progiciel aussi critique qu'un ERP.

Ces diverses solutions pèchent souvent par:

- Une trop faible communauté
- Une plateforme technique en bêta de l'aveu même des concepteurs
- Un fonctionnel encore trop limité
- Une francisation trop peu aboutie; on est en général alors loin d'un comptabilité française et on ne parle ici que de francisation balbutiante de l'interface

Nous souhaitons bonne chance à ces nouvelles initiatives et les garderons en ligne de mire dans notre activité de veille.

Toujours pour éviter les risques, nous rejetons tout ERP se disant libre mais n'ayant pas de **licence standard reconnue par l'OSI** (Open Source Initiative) et qui permettrait à l'éditeur de rendre ensuite ses utilisateurs captifs d'une solution qu'il serait le seul à dominer.

Nous rejetons les ERP n'ayant pas d'**envergure internationale**. Un ERP est un produit si complexe qu'il faut une grande base d'utilisateurs/contributeurs pour le rendre fiable et attractif. Ainsi les ERP qui n'ont pas au moins une documentation centrale en anglais sont rejetés.

ERP open source

Nous rejetons les ERP qui n'a pas de **forum** de discussion librement accessible ou une **mailing liste active**. C'est le minimum de transparence à exiger pour pouvoir évaluer un produit.

Architecture technique minimale

Passons en revue les exigences minimales au plan technique:

- Au niveau du langage:
 - Rejet des langages dont le **modèle objet** n'est pas assez fiable et ne garantissant pas ainsi un minimum de règles telles que: l'encapsulation et la factorisation du code: ainsi PHP4 est rejeté (PHP5 aurait été accepté)
 - Rejet des langages ne garantissant pas une bonne **lisibilité** et **maintenabilité** du code: PERL est rejeté par exemple
 - Rejet des langages imposant aux développeurs la responsabilité de la **gestion de la mémoire**. En effet, hormis pour le SGBD, un ERP doit privilégier les fonctionnalités, la maintenabilité et la fiabilité sur la vitesse d'exécution pure: ainsi C et C++ comme langages d'implémentation fonctionnelle sont rejetés.
- Exigence d'un découpage clair du code et de ses responsabilités, notamment **MVC**.
- Solution standard de stockage de données, en l'occurrence une **base relationnelle connue et fiable**. Ceci nous pousse à éliminer ERP5 à cause de ses bases objet Zope. La solution de Zope a beau être valide, son problème est d'être peu connue.
- Afin de faciliter la lecture du code, sa réutilisation et l'**encapsulation** de ses responsabilités, **une couche d'accès aux données est requise**. Cette couche, ou ORM, doit permettre de manipuler les entités relationnelles en terme d'objet dont le niveau d'abstraction est plus fort et plus proche du fonctionnel métier. Hélas, les ORM rencontrés ne sont pas tous de bonne qualité et il faut parfois composer avec des vestiges de code purement relationnel.
- Pour des raisons de maintenabilité évidentes, les très nombreuses vues (au sens du modèle MVC) d'un ERP ne peuvent être entièrement codées à la main en langage de bas niveau par exemple proche du HTML. Heureusement, dans les solutions restantes, aucune n'est tombée dans cet écueil et toutes adoptent des solutions de **méta-programmation** ou de génération de code.

Sélection des ERP open source satisfaisant à nos critères

Voici un récapitulatif de notre passage au crible exhaustif des ERP's open source. Bien sûr en premier approche nous avons dû être très sévère par rapport à notre critères de filtrage, sachant qu'au final il nous fallait une liste suffisamment petite pour qu'on puisse l'étudier. Pour chaque solution, on a éventuellement grisé les cases où un de nos critères a été éliminatoire.

ERP open source

Solution	techno	périmètre fonctionnel	qualité technique	références client connues	Notoriété web PageRank	notoriété web Delicious	posts/mois sur forum+mailing	Dynamique Google Trend	communauté	gouvernance	web?	compa française	licence	critère d'élimination éventuel
OFBiz	J2EE	+++	++	Yves Rocher Messer Group labellecolec.fr	7	425		légère baisse	+++	-- (pas de business model théorisé)	OUI	--	Apache2	Ofbiz n'est qu'un framework, Neogia préféré
Neogia	J2EE	+++	+++	Messer Group	5	17	10 non suivi!		++	-?	OUI	++	Apache2	
OpenTaps	J2EE	+++	?	?	5,5	234		néant	+++	++ ? base Open	OUI	--	HPL	
Compiere	J2EE	+++	+	Goodyear Germany, Leplombfrancais	6	209	100 baisse		--	Source, plus de dynamique d'intégrateurs autogérée, pas de leader	OUI	+++	(~ GPL 3 Aferro)	comme Neogia en moins bien
Adempiere	J2EE	+++	+		5	108	500 stable		+++	+++ (business model théorisé) par l'éditeur	OUI	+	GPL -> 2006	
OpenBravo	J2EE	+++	+		6	853	1000 explosion		+++	+++ (business model théorisé) par l'éditeur	OUI	+	GPL	
OpenBlueLab	J2EE	?	?		5	11		néant	?	?	?		-MPL	
Plazma ERP	J2EE	+	+++		5	14	0 néant				NON	--	LGPL	pas très mature, pas de gestion de production
JFire	J2EE	?	+++		0	100		néant	?	?	partiel	--	LGPL	statut beta au 12/2007, pas prêt d'être francisé
JallInOne	J2EE	+++			2	10	5				NON	--		pas assez connu pas web
OpenAguila	J2EE	+	?		4	50		néant	-	?	OUI	++++		diffusion trop confidentiel

ERP open source

lution	techno	périmètre fonctionnel	qualité technique	références client connues	Notoriété web PageRank	sauvé sur sur Delicious	posts/mois sur forum+mailing	Dynamique Google Trend	communauté	gouvernance	web?	compta française	license	critère d'élimination éventuel
nyERP	PostgreSQL XMLRPC SOAP	++++	++++	Whirlpool, Hotels Costes, IRMiéro systems	6	117	500 explosion	500 explosion	+++	++++	OUI	+++	GPL	
P5	Python/Zope/B ase Objet	+++	+++	Adia Senegal, Infoterra, Corami, Astrium	5,5	177		néant	+ ?	+ ?	OUI	+++		ij pas explosion, base zope+python => trop exotique
QUEnterprise	python/GTK	?	-- ?		6	123		néant	++	?	NON	--	GPL	pas assez connu pas web
libarr	PHP4	++ ?	---		5	15		néant	++	+ ?	OUI	--		orienté TPE, PHP4!
enPro	PHP4				5	48		néant				--		
ibERP	PHP4	?	?		0	181		néant	?	?	OUI	--		pas français, PHP4
ject-open	PHP4 (eGroupware)	+++			5	375		néant						
te-erp	PHP4 (eGroupware)	+++			4,5	15		néant						
erp	PHP4				4	17		néant						
jecterp	PHP5				4	0		néant			OUI			

ERP open source

nom	techno	périmètre fonctionnel	qualité technique	références client connues	Notoriété web PageRank	sauvé sur sur Delicious	posts/mois sur forum+mailing	Dynamique Google Trend	communauté	gouvernance	web?	compta française	license	critère d'élimination éventuel
ce	Ruby on Rails	++	++++		3	78		néant	-		OUI			pas assez connu pas assez mature
llo-cmms-os	Ruby on Rails	++	++++		2	10		néant			OUI			pas assez connu pas assez mature
edger books dérivé	PERL	++	?		6	280		néant	?	?	?		GPL	PERL, faible doc
ercial: openmfg/natif/GTK/Post mais xTuple	natif/natif/natif/GTK/Post greSQL	?	?		3	21		néant	?	?	NON			
Cash	natif/GTK	--			7	1551		néant			NON	--	GPL	orienté compta seule et TPE
ra	natif/GTK/Post greSQL				4	25		néant	- (ES only)		NON	--	GPL	
q	natif/QT				5	29		néant	- (ES only)		NON	--		

Établissement d'une 'shortlist'

Sur la base des critères évoqués ci-avant, le tableau récapitulatif explique le rejet d'un bon nombre de solutions.

Finalement, les ERP à considérer pour un usage immédiat dans le contexte du marché français se résument à :

- TinyERP
- Openbravo
- Adempiere
- Compiere
- ERP5
- Neogia

Notons que OfBiz et Opentaps sont très proches de Neogia, lequel en dérive directement. En ce qui concerne Open For Business (OfBiz), comme nous l'avons déjà dit, **il s'agit plus d'un framework**, qui privilégie l'évolution du code sur la stabilité et en aucun cas d'un ERP clé en main. Seuls Opentaps et Neogia qui en dérivent directement ont l'ambition d'être de tels ERP assez directement utilisables. Mais Opentaps n'est pas francisé de façon satisfaisante. Dès lors, dans cette famille, autant retenir Neogia qui importe également d'Opentaps ce qu'il manque à OfBiz

ERP open source

SÉLECTION RÉDUITE DES MEILLEURS ERP

Profil macroscopique des ERP pré-sélectionnés

Le tableau de données précédant, s'il a été utile au filtrage, n'offre pas de lisibilité macroscopique. Nous définissons donc d'agréger, certes plus subjectivement, un certain nombre d'indicateurs offrant cette lisibilité.

Bien sûr, votre choix final devra tenir compte de l'adéquation des fonctionnalités de l'ERP, surtout si c'est la compétitivité native que vous recherchez. Or l'étude des fonctionnalités des ERP retenus est très complexe et a donc été développée spécifiquement (voire les chapitres précédents). Nous offrons donc ce profil macro sans entrer dans le détail des fonctionnalités, en considérant uniquement la couverture globale regroupé sous l'axe 'périmètre'.

Profil par caractéristiques générales

Notoriété actuelle

La notoriété actuelle d'une solution est importante dans la mesure où elle est source de sécurité ou au moins de précaution. Cependant, une notoriété construite au moyen d'investissements marketing ne tiendra pas longtemps si elle ne va pas de pair avec dans une dynamique communautaire et une qualité technique de fond.

S'il est déterminant d'implémenter une bonne solution à l'instant t, il l'est tout autant de s'assurer que les indicateurs sont bien au vert pour que la solution reste bonne dans 5 ans au moins. En effet, disponibilité des ressources, prix et possibilité des évolutions vers des verticalisations en dépendront directement. Voilà pourquoi ce critère de notoriété, s'il est important n'est pas suffisant. Reste que Compiere se distingue ici par plusieurs belles références sur le marché français.

ERP open source

Sont considérés:

- Nombre et importances des références client
- Nombre et notoriété des intégrateurs existants (s'agit-il uniquement d'amateurs isolés ou de vraies entreprises ? N'y a t-il qu'un seul intégrateur derrière un produit ?)
- Citations dans la presse professionnelle
- Taille des archives des forums et mailing lists
- PageRank Google du site, lié au nombre de sites eux même importants pointant sur le site de l'ERP
- Bookmark d'internautes. Ici c'est Del.icio.us (système Yahoo) qui a été utilisé comme étalon de mesure. Moins de 20 bookmarks du site institutionnel de l'ERP là où certains ERP cumulent 800 bookmarks (Openbravo) signifie vraiment qu'il y a un souci sur la masse critique de la communauté de développeurs. Neogia était à la limite. Cependant nous le retenons car Neogia est avant tout un portage d'Ofbiz dont l'activité communautaire est indéniable.

Dynamique

Il s'agit de la dynamique communautaire autour de la solution. Avec la qualité technique, elle va déterminer directement la place de la solution dans le futur. En effet, les investissements d'un éditeur sont finalement peu de choses au regard de tous les correctifs, toute la documentation et même tout le marketing qui est fait par une communauté active.

Ce qu'on peut ici dire, c'est que le *business model* où l'éditeur est son seul intégrateur est éliminatoire, et que le nombre très fermé d'intégrateurs choisis par Compiere et ERP5 nuit clairement à leur dynamique. De, même, en dépit de l'activité sur OfBiz, Neogia, semble avoir marqué un net ralentissement de son activité de développement courant 2007. Est-ce dû à la faiblesse relative de son unique intégrateur français Nereide ?

Sont considérés:

- Évolution du volume de forums ou mailing list
- Activité des chats irc
- Gouvernance: dans quelle mesure intégrateurs et utilisateurs sont ils consultés et partie prenante dans la conception et l'évolution du produit?
- Fréquence des mises à jour de la documentation, notamment des wiki
- Activité du dépôt de code CVS/SVN
- Fréquence des releases

ERP open source

- Citations par des acteurs indépendants dont les bloggers occasionnels (qui sont en fait une armée 'invisible' de correcteurs potentiels de bug ou auteur de documentation).

Techno

Investissements et communauté sont encore peu de chose devant la cohérence, la puissance et l'adéquation avec les standards des modélisations au cœur d'un ERP.

Les fonctionnalités ne sont qu'une surcouche à ces fondations et le coût d'implémentation d'une même fonction métier varie très facilement de 1 à 5 selon la qualité technique des fondations. Ainsi, au delà d'un certain stade, même à grand renforts d'investissements, on ne peut plus enrichir le fonctionnel d'un ERP s'il est basé sur des abstractions de trop bas niveau alors qu'une solution bien architecturée faite de concepts clairs et efficaces pourra au contraire s'étendre à moindre frais. Bien sûr, un éditeur en bonne santé financière peu recoder sa solution - et c'est courant - mais, vous, serez surtout lié à l'ancien produit avec lequel il aura fallu faire rupture. Si une solution ne peut pas être améliorée de façon continue, c'est vous qui ferez les frais des migrations brutales. Et plus directement, le coût de vos développements spécifiques dépend de la qualité technologique. Les ERP de la lignée Compiere qui ont une large base de code en pl/SQL sont ici moins bien positionnés.

Sont considérés:

- Respect de standards existants si possible (gage de maintenabilité et de facilité de prise en main)
- Puissance et canonicité des abstractions mises en jeu (gage de productivité; sous entendu ici: ORM, moteur BPM, webservices natifs...)
- Degré de factorisation du code (gage de fiabilité et de prise en main)
- Habileté des 'hooks', points d'ancrage, et interfaces pour les extensions spécifiques
- Maturité et couverture des webservices
- Courbe d'apprentissage du produit: une courbe plate est moins bien notée
- Modularité de l'application (pattern *Inversion Of Control* si possible afin que l'application soit composée d'un noyau minimal et de plugins qui sachent bien tenir compte les uns des autres)
- Absence de problème évident de performance

ERP open source

Périmètre

Il s'agit ici du volume global des fonctionnalités. Bien sûr, **il faudra vérifier le détail** dans les chapitres dédiés à chaque ERP car aucune confrontation macroscopique n'est possible.

A noter qu'il faut aussi garder cette vision globale des fonctionnalités. Beaucoup de fonctionnalités de niche ne seront jamais utilisées ou devront être modifiées. Le critère de 'souplesse' est autrement plus impactant en termes de coût ou de capacité à coût donné. D'autant que sur un ERP souple, l'ajout d'une fonctionnalité peut se révéler relativement simple. A un stade macro, on retiendra par exemple que Compiere est plus pointu sur le CRM qu'Openbravo alors que ce dernier est meilleurs dans la gestion de production. En revanche, le détail des fonctions CRM par exemple est très peu discriminant compte tenu des facultés d'adaptation. TinyERP se distingue ici avec son activité communautaire très importante et ses plus de 200 modules.

Souplesse

Dans la mesure où on doit très souvent dépasser le périmètre fonctionnel natif de l'outil, quelle facilité a-t-on à le faire? Il s'agit donc d'un **critère absolument déterminant dans le coût total de possession** compte-tenu du fort coût relatif des développements spécifiques. La souplesse rejoint ici largement la technologie mais elle met spécifiquement l'accent sur la modularité de la plateforme de l'ERP et sur l'efficacité du développement par des tierces parties.

Sont considérés:

- Facilité à modifier les structures de données pour ajouter ou altérer le stockage des objets métier
- Facilité à modifier les interfaces utilisateurs pour leur donner une bonne ergonomie métier
- Présence ou non d'un moteur de workflow et puissance de celui-ci le cas échéant. Est-il graphique ou faut-il coder du XML?
- Facilité à altérer non pas les structures de donnée mais les traitements à effectuer. Les objets métier peuvent-ils être des vrais objets conceptuels qui ont des méthodes attachées ou bien faut-il au contraire associer des procédures à des objets avec donc moins de maintenabilité et d'agilité?
- Facilité à adapter les rapports (factures et autres)
- Rapidité, légèreté, des cycles de développement: faut-il recompiler, re-déployer des classes, importer ou exporter des méta-données dans la base de données, dès lors comment déploie-t-on des adaptations fonctionnelles sur une base de production?

ERP open source

Ressources

Difficulté ou non, sur le marché français, à mobiliser des ressources capables d'effectuer des développements pointus sur l'outil. Ceci est assez discriminant pour ERP5.

Attention néanmoins à ne pas mal interpréter cet indicateur car une technologie meilleure peut largement rentabiliser un effort d'adaptation initial supérieur. C'est pour ceci que nous n'avons pas ici trop pénalisé TinyERP. En effet, son code clair d'un haut niveau d'abstraction (proche des spécifications fonctionnelles) sera facilement maîtrisé, y compris par des codeurs ne dominant pas le Python, langage moins connu en France que Java par exemple.

Des ERP tels que TinyERP ont une très bonne capacité à être configurés (notamment workflows, rapports) et requièrent donc moins de développement spécifique.

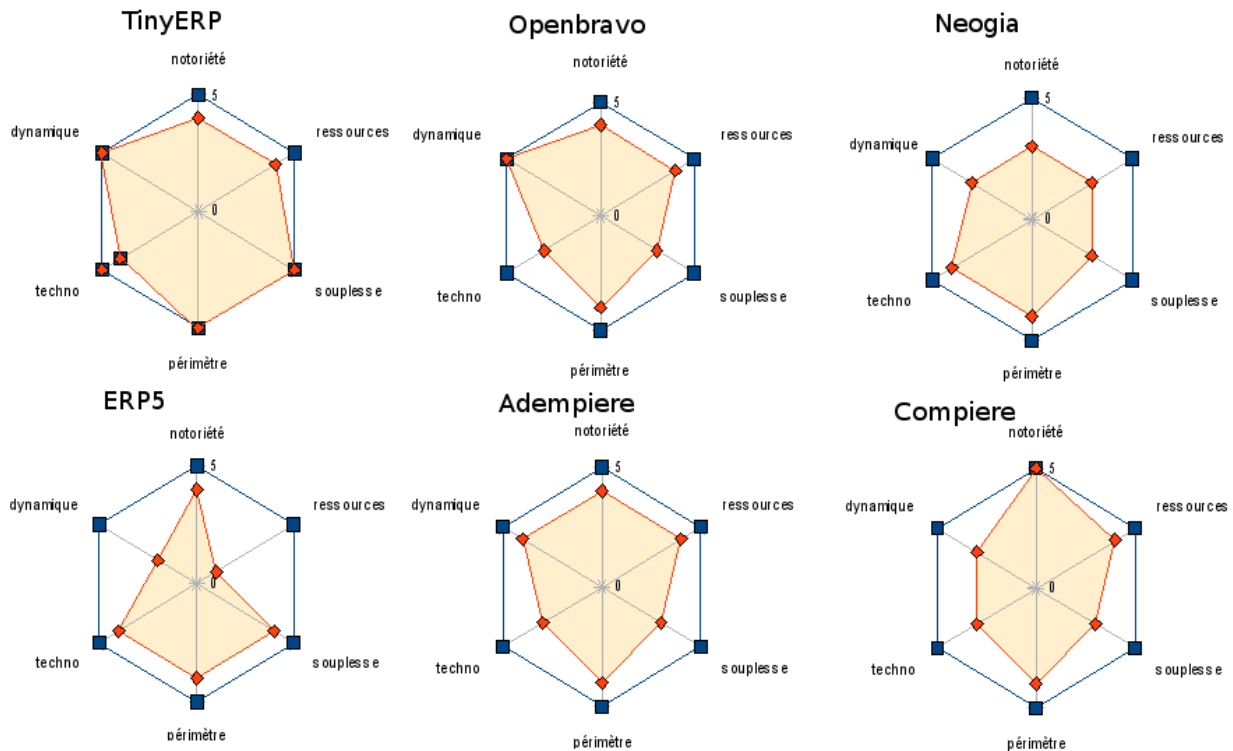
Conclusion

Au final, on peut synthétiser avec le tableau et le graphe suivant:

	notoriété	dynamique	techno	périmètre	souplesse	ressources
TinyERP	4	5	4	5	5	4
OpenBravo	4	5	3	4	3	4
Neogia	3	3	4	4	3	3
ERP5	4	2	4	4	4	1
Adempiere	4	4	3	4	3	4
Compiere GPL	5	3	3	4	3	4

Toutes évaluation entre 0 (faible) et 5 (excellent)

ERP open source



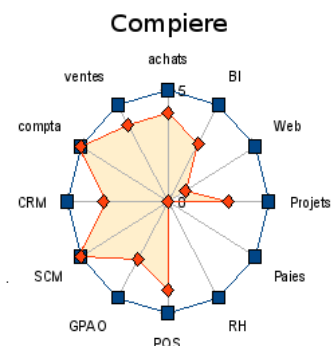
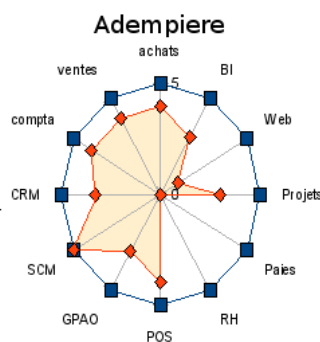
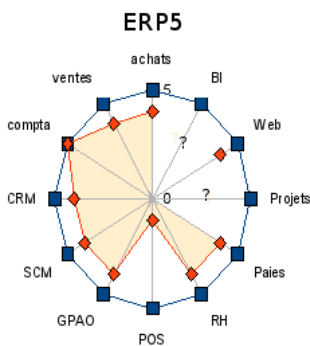
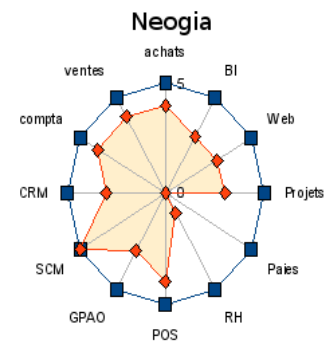
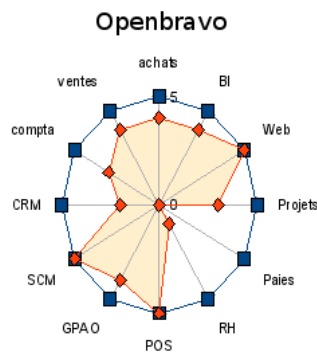
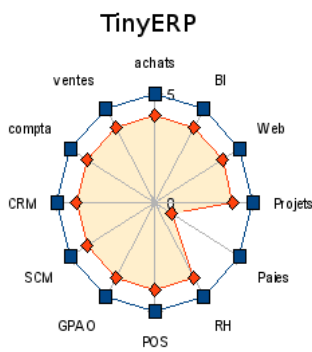
Profil par domaine fonctionnel

Voici un récapitulatif des capacités **relatives** – mais non proportionnelles – de chacun des ERP retenus sur les domaines fonctionnels les plus caractéristiques (de 0 à 5 pour le plus adapté). Les différences les plus flagrantes se font sentir sur les modules de gestion des ressources humaines pour lequel seuls ERP5 et TinyERP sont complets. ERP5 va même jusqu'à gérer les paies alors qu'aucun autre ERP libre n'est allé aussi loin (TinyERP s'y essaie et dispose d'un module de paie suisse par exemple, mais pas français). Sans module RH, la gestion de projet est aussi plus limitée et c'est ainsi que TinyERP traite mieux que ses concurrents ce domaine fonctionnel. De même, ERP5 et TinyERP sont plus complets sur la CRM, où Openbravo est plus limité. En revanche, ce dernier se distingue avec son interface web inégalée et son module point de vente avancé.

ERP open source

	achats	ventes	compta	CRM	SCM	GPAO	POS	RH	Paies	Projets	Web	BI
TinyERP	4	4	4	4	4	4	4	4	1	4	4	4
OpenBravo	4	4	3	2	5	4	5	0	0	3	5	4
Neogia	4	4	4	3	5	3	4	1	0	3	3	3
ERP5	4	4	5	4	4	4	1	4	4	?	4	?
Adempiere	4	4	4	3	5	3	4	0	0	3	1	3
Compiere GPL	4	4	5	3	5	3	4	0	0	3	1	3

Toutes évaluation entre 0 (faible) et 5 (excellent)



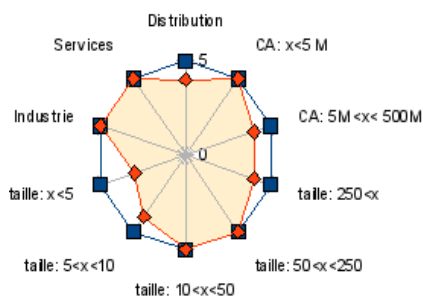
Profil par secteur et taille d'entreprise

Si on considère les aptitudes fonctionnelles particulières des ERP retenus ainsi que leurs autres caractéristiques comme le degré d'ouverture de l'éditeur, la qualité de finition, la complexité ou simplicité des extensions, on peut déduire un profil type d'utilisation, selon le secteur d'activité, mais aussi selon la taille et le chiffre d'affaire de la société. En voici un tableau récapitulatif (de 0 à 5 pour le plus adapté).

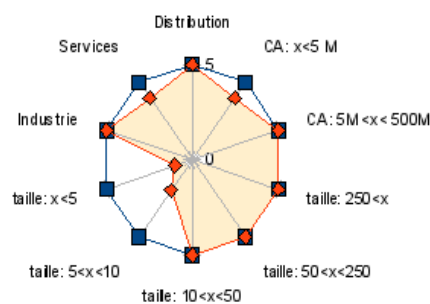
ERP open source

	secteur			taille de l'entreprise						CA	
	Distribution	Services	Industrie	x<5	5<x<10	10<x<50	50<x<250	250<x	x<5 M	5M <x< 500M	
TinyERP	4	5	5	3	4	5	5	4	4	5	
OpenBravo	5	4	5	1	2	5	5	5	5	4	
Neogia	5	4	4	0	2	3	3	4	4	3	
ERP5	4	4	4	3	4	5	4		4	4	
Adempiere	5	3	3	1	2	5	5	3	4	3	
Compiere GPL	5	4	3	2	2	5	5	5	5	3	

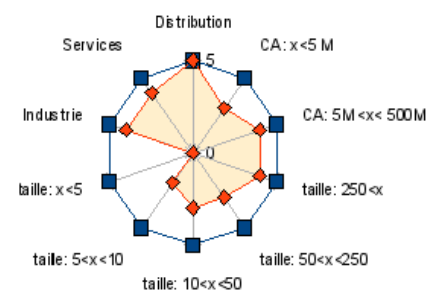
TinyERP



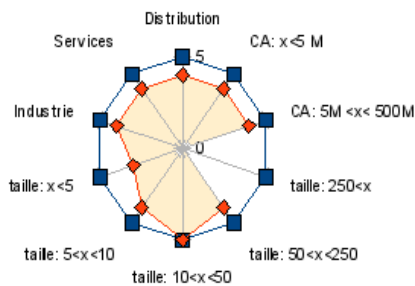
Openbravo



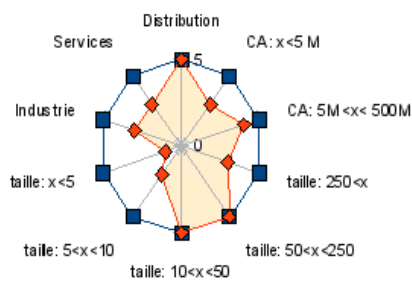
Neogia



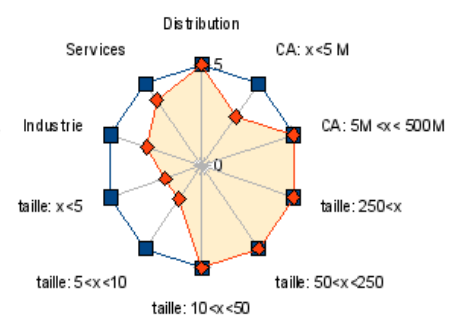
ERP5



Adempiere



Compiere



Secteur d'activité

Tous les secteurs d'activité n'ont pas les mêmes besoins et tous les besoins ne sont pas aussi complexes.

Distribution/Logistique

La distribution aura surtout besoin d'une bonne gestion d'inventaire, de politique de prix fine, d'interfaces par webservices avec d'autres applications telles que des plateformes de vente e-commerce. Globalement, ces besoins sont assez simples et standards.

ERP open source

Ainsi, tous les ERP retenus y sont globalement adaptés. On trouvera un moteur de règle de pricing très similaire dans Compiere, Adempiere, Openbravo et TinyERP par exemple. Ces derniers et Neogia gèrent les inventaires avec une comptabilité à double entrée ce qui est un gage de sécurité (enregistrement des mouvements et pas seulement des quantités des stocks).

De même, les ERP retenus ici gèrent tous les stocks multiples et même la consolidation hiérarchique des stocks pour ce qui concerne TinyERP ou Neogia: on peut connaître un stock de pièce d'un entrepôt globalement même si celui-ci est par exemple décomposé en deux parties par exemple pour matérialiser le passage ou non d'un contrôle qualité dans un même entrepôt physique.

On notera quand même que la distribution peut brasser d'importants **volume de ventes** et qu'à ce titre, ERP5, avec sa base de données qui n'est pas un standard connu peut donner lieu à quelques doutes en terme de tenu en charge. A ce jeu, c'est sans doute Openbravo, Compiere ou Adempiere qui l'emporteront avec leur capacité à être utiliser optionnellement avec une base Oracle les performances sont rassurantes. En tout état de cause, à partir de plus de 10 000 mouvements de stock par jour environ, il faudra étudier sérieusement la faisabilité par rapport à l'ERP retenu.

Enfin, certains ERP se sont aussi spécialisés dans des solutions de point de vente. C'est typiquement le cas d'Openbravo dont l'éditeur a intégré nativement la solution de vente en caisse réputée et open source TinaPOS, depuis repackagée et renommée OpenBravoPOS. TinyERP possède lui aussi un module de point de ventes spécialisé. Enfin Compiere et Adempiere peuvent aussi s'interfacer avec des solutions open source dont TinaPOS ou ses forks à des degrés plus ou moins natifs. Quant à Neogia, il semble qu'il soit plus difficile de lui adjoindre une interface de vente en caisse aboutie en dépit de l'existence de quelques prototypes hérités d'Ofbiz.

Services

La gestion des sociétés de services est déjà un peu plus complexe. On a toujours les même notions de règles de tarification, mais on doit surtout budgéter des projets et comparer ces budgets aux ressources humaines réellement consommées par ces projets. Les projets doivent être découpés en lots ou en tâches avec une gestion des dépendances entre les tâches. Pour bien fonctionner, l'affectation des ressources humaines aux tâches suppose un certain niveau de gestion des ressources humaines. Finalement, une série d'outils sert à monitorer et à piloter les projets selon des méthodologies aussi diverses qu'on peut imaginer de verticalisations sectorielles.

Dans ce secteur, la famille des Compiere (Openbravo et Adempiere inclus) n'est pas très à l'aise à cause du manque de gestion des ressources humaines essentiellement. Openbravo et Compiere permettent, par exemple de créer des budgets pour des projets à facturer, puis d'imputer

ERP open source

globalement le coût effectif du projet afin de permettre le suivi des coûts. Il manque cependant les aspects de gestion de dépendances entre tâches, leur planification et leur affectation.

En revanche, TinyERP sort nettement du lot pour ce secteur. Il permet en effet de gérer nativement les ressources humaines, les planning, demandes de congés et beaucoup d'autres contraintes liées aux RH. De plus, ses '*dashboard*', tableaux de bord personnalisés permettent d'offrir au chef de projet une surveillance personnalisée de ses projets (taux d'avancement, de rentabilité etc...) alors que les autres employés peuvent effectuer leurs imputations réelles sur les tâches qu'on leur a assignées et ce, sans aucune assistance.

Grâce à sa modularité incomparable, TinyERP offre déjà des modules de gestion de projet verticalisés tel que la gestion de projet informatique agile grâce à son module **SCRUM** par exemple. Il ne tient qu'à vous d'ajouter relativement facilement une gestion de projet propre à votre activité spécifique (moins de 500 lignes de code, surtout déclaratif, pour la gestion SCRUM, Model, Vues et Controllers compris).

Ajoutons que pour le cas précis de TinyERP, sa fonctionnalité de **BPM** (*Business Process Management*) intégrée lui permet d'affronter, sans écriture de code, les workflows organisationnels les plus complexes et ainsi d'offrir une grande souplesse organisationnelle aux sociétés de services.

Bien que TinyERP s'enrichisse à une grande vitesse, il faut aussi noter que début 2008, la gestion des projets reste néanmoins inférieure à ce que fait par exemple **Microsoft Project**. En effet, s'il est bien possible d'afficher les diagrammes **GANTT** des tâches par exemple, les dépendances entre tâche n'étaient toujours pas gérées. Ainsi la planification des projets complexes à tâches parallélisables ou non ne pouvait pas avoir lieu automatiquement; il fallait en passer par une planification manuelle sur ces projets. L'éditeur de TinyERP a néanmoins annoncé des développements en ce sens.

Industrie

Au delà des contraintes déjà citées, l'industrie se caractérise par la production de produits. Cette production doit être gérée. Dans les cas les plus simples, on ne fera que gérer dans l'ERP des stocks dans différents entrepôts puis on mettra à jour ces stocks manuellement en fonction des opérations de production. On utilisera aussi l'ERP comme référentiel de stockage des opérations de production (les ordres de production). Tous les ERP retenus permettent un tel fonctionnement basique.

Cependant, les besoins vont généralement beaucoup plus loin que l'aspect référentiel et traitements manuels. Dès que l'on monte en complexité on exige la gestion des nomenclatures et des gammes.

A partir de là, seuls TinyERP, Openbravo, ERP5 et Neogia restent en course. Si la MRP vient d'être ajoutée sur Compiere, nous ne pouvons pas

ERP open source

encore estimer qu'elle soit mature. Et, bien que ceci puisse venir d'un manque d'approfondissement, nous exprimons ici aussi des doutes sur les capacités réelles génériques d'ERP5 et de Neogia sur ces points. Il manque encore une masse critique de retour d'expérience pour valider leur gestion de production en dehors des rares *success stories* qui ont impliqué l'éditeur directement. Ainsi nous pensons que seuls tinyERP et Openbravo peuvent être sérieusement envisagés pour la MRP1.

Pour des besoins plus haut de gamme, une entreprise industrielle sera à la recherche de solutions de MRP2, c'est-à-dire, non pas seulement de simple calcul des besoins étant donné les prévisions de vente, mais aussi d'une planification de cette production (et des achats d'approvisionnement). C'est bien là actuellement la limite des ERP libres, ils n'excellent pas dans ce domaine très spécialisé de recherche opérationnelle et de programmation par contraintes comme nous l'avons déjà évoqué, mais peuvent toujours importer une planification réalisée par un outil externe comme on l'a déjà souligné.

Taille de l'entreprise et chiffre d'affaire

C'est sans doute le plus subjectif de nos critères. Il s'agissait néanmoins pour nous de soulever quelques différences entre les solutions retenues.

Limite haute effectif et chiffre d'affaire

En fait, il n'y a pas de vraie limite haute avérée. Disons plutôt que nous disposons d'un certain volume de retours d'expériences sur des gros déploiements jusqu'à une certaine taille et pas encore au delà. L'ERP étant un progiciel vital pour l'entreprise il convient d'étendre cette couverture par itérations successives et petits sauts afin de ne pas prendre le risque de rater un projet. Notons aussi qu'un cas unique de déploiement n'est pas très parlant, les conditions de sollicitation et de complexité peuvent varier énormément.

Limite basse effectif et chiffre d'affaire:

En revanche cette limite ci est plus évidente. Les petites PME et TPE seront très intéressées par la scalabilité vers le bas de certains ERP. En effet, une TPE peut-elle assumer le prix de la mise en place et de la maintenance d'un pont comptable entre l'ERP et un Sage ou un Ciel? Nous pensons que ça sera rarement le cas et donc qu'à ce titre Openbravo, par exemple, sera inadapté aux TPE françaises au moins jusqu'à ce que la comptabilité native d'Openbravo voit sa francisation reconnue. Pour TinyERP, ERP5 ou Compiere, une TPE courageuse et compétente pourra néanmoins utiliser la comptabilité générale de l'ERP à condition d'y dédier le temps requis.

De même, une TPE, pourra-t-elle, envisager de payer des développements spécifiques? Là encore nous pensons que non et que l'ERP à déployer doit être paramétrable et maintenable sans compétence de développeur et qu'il doit être déjà **verticalisé au plus proche du cas d'usage**. A ce titre là,

ERP open source

nous pensons qu'ERP5 et TinyERP, parce qu'ils sont basés sur des technologies plus modulaires et plus souples (BPM, code objet), sont plus adaptés aux petites entreprises. Attention, cela ne veut pas dire qu'il le soit moins pour les grosses.

De plus, les TPE qui ne veulent pas investir dans des intégrations de développements spécifiques ni dans la maintenance de leur serveur pourront investir dans les **solutions hébergées** lancées par ERP5 (ERP5Express) et TinyERP (OpenERP OnDemand) qui sont très compétitives en terme de prix et proposent d'utiliser le même ERP libre que vous pourriez télécharger et héberger vous même en cas de besoin. Leur capacité d'intégration devra néanmoins être vérifiée.

Choix entre Compiere, Adempiere et Openbravo

Comme nous l'avons vu, Compiere, Adempiere et Openbravo sont assez proches, tant en terme de technologie que de fonctionnalités actuelles.

Cela ne doit pourtant pas masquer des différences tout à fait essentielles:

Des écarts fonctionnels réels

Malgré leurs similitudes, il existe des différences fonctionnelles qui pourront faire la différence sur des cas d'intégration particuliers:

- Openbravo est plus abouti en Gestion de Production et sera donc plus adapté aux industries que Compiere.
- Openbravo, par son association avec TinaPOS devenue Openbravo POS va aussi beaucoup plus loin dans la gestion des points de vente, ce qui intéressera particulièrement le secteur de la distribution.
- Compiere est, en revanche, légèrement plus avancé sur la CRM.
- Compiere possède aussi une comptabilité générale plus proche de ce qui est exigé en France si on veut que l'ERP gère la comptabilité générale (cf notre section sur la comptabilité page 50)
- Adempiere ressemble à Compiere modulo quelques avancées communautaires (par exemple sur la gestion de l'EDI) mais sera moins fiable sur la comptabilité française justement.

L'accès web discriminant

Comme nous l'avons déjà évoqué, Openbravo a fait le choix d'un accès web depuis son fork en 2002. Ainsi, Openbravo offre un accès web à l'ensemble de ses fonctionnalités et ce gratuitement et sous licence open source (MPL).

ERP open source

Au contraire, sur Compiere et Adempiere, le support du web est assez récent. A l'exception du travail encore peu fiable de quelques intégrateurs spécialisés, Adempiere n'a pas encore su prendre ce virage. Du côté de Compiere, le virage web a enfin été réalisé et de belle manière avec le framework GWT de Google. Cependant Compiere ne le distribue pas sous licence open source et encore moins gratuitement ('Compiere Professionnal' coûte ainsi 500 euros par mois pour une base de 10 utilisateurs selon le prix catalogue constaté début 2008).

Des différences d'ordre politique

Jusqu'où va l'ouverture de Compiere?

Si on a entendu parler d'un ERP libre, c'est généralement de Compiere. Et pour cause: Compiere était un pionner dans les années 2002.

Hélas, depuis 2006, en cherchant à rentabiliser une levée de fonds récente de manière indélicate pour sa communauté, Compiere Inc s'est brusquement coupée de l'essentiel de sa base communautaire.

Suite à cet imbroglio de gouvernance en 2006¹¹, de nombreux forks de Compiere ont été créés. Aujourd'hui, c'est assurément Adempiere qui fédère derrière lui ce qui subsiste de la communauté Compiere.

En effet, les griefs contre Compiere sont:

- Compiere fixe trop cher le ticket d'entrée pour se former à son outil, ne serait-ce que pour l'évaluer (toute documentation est payante).
- Compiere ne maintient aucun espace communautaire sur lequel un écosystème d'intégrateurs/contributeurs puisse continuer à se développer
- Compiere rend difficile la lecture de l'évolution du code ainsi que de sa 'roadmap', il ne cherche pas à tenir compte des avis d'une communauté de développeurs et d'utilisateurs autres que ses clients directs.

En conséquence, l'écosystème Compiere est assez inerte. Bien que la licence de Compiere soit open source (GPL2), nous estimons que Compiere n'est pas assez ouvert aux contributions externes. Soyons plus concret. C'est tout le contraire de TinyERP, par exemple, qui a su construire un modèle plus ouvert, où de très nombreux modules sont développés par des entités externes à l'éditeur. Ainsi, le module pour l'interface web, eTiny a été développé par un intégrateur (Axelor) puis reversé à la communauté. Axelor en tire bien entendu un bénéfice en termes d'image et les utilisateurs finaux bénéficient gratuitement du client web sur TinyERP. A l'inverse, dans Compiere, le client web est développé par Compiere Inc et est payant.

¹¹<http://red1.org/forum/viewtopic.php?t=931>

ERP open source

Au final, Compiere a plusieurs handicaps :

- Pas assez bon en gestion de production
- Peu adapté aux sociétés de service
- Client trop lourd pas assez souple aux développements spécifiques faits sur le serveur.
- Communauté trop inerte (pas plus de 100 posts par mois sur les forums de discussion contre 1000 pour Openbravo ou 500 pour TinyERP).
- Évolution fonctionnelle trop lente
- Interface web ni open source, ni gratuite
- Risque: en l'absence de communauté: si Compiere décide de ne plus sortir de nouvelles versions libres mais seulement des upgrades/patches payants, personne d'autre ne le fera à leur place et dans les faits vous serez captif de l'éditeur et de sa politique tarifaire pratiquement au même titre qu'une solution propriétaire.

Même si Compiere reste intéressant en dépit de ces notes négatives, nous estimons qu'il y a des concurrents plus avantageux que Compiere dans la plupart des situations. Dès lors mieux vaut partir avec eux. Nous verrons bien si Compiere sait renouer avec une communauté ou quel effet l'absence de communauté aura sur la pérennité du produit à long terme. En revanche, si Compiere 'out of the box' vous satisfait en grande partie, que vous avez le besoin de son moteur de workflow minimaliste ou de sa gestion comptable native, alors pourquoi pas.

Adempiere, quelle fiabilité?

Comme nous l'avons dit, Adempiere est né du fork de Compiere en 2006 pour contrecarrer la politique trop unilatérale de Compiere Inc vis à vis de sa communauté. Depuis Adempiere a su fédérer tous les forks récents de Compiere derrière lui.

Cependant, si on regarde les évolutions d'Adempiere jusqu'à maintenant, on s'aperçoit qu'il s'agit essentiellement de

- Correctifs de bugs par rapport à la version forkée de Compiere
- Très nombreux développements spécifiques d'extension mais qui ne sont pas ou mal intégrés à Adempiere. En fait il s'agit de modules phares vitrine pour les intégrateurs plus que de véritables enrichissements cohérents du produit.

Il y a bien une roadmap assez rassurante avec notamment:

- la volonté de développement d'une interface web

ERP open source

- la volonté de passer à de meilleures abstractions au niveau de l'ORM, notamment en utilisant le framework réputé Hibernate.

Le gros souci vient du fait que **depuis le fork aucun développement d'une telle envergure n'a été mené**. Dès lors il est permis de douter de la capacité actuelle d'Adempiere à réaliser ces développements.

A l'heure actuelle, Adempiere ne saurait donc pas non plus rassurer sur son futur et donc dans ces conditions, nous prônerions plutôt la circonspection en attendant que la gouvernance imaginée par la communauté fasse ses preuves.

Openbravo la solution ?

Openbravo, au contraire semble mettre la barre plus haut que Compiere en terme d'ouverture et s'inscrit dans une démarche de conquête plus agressive.

En effet, Openbravo investit clairement dans une activité communautaire importante: là où Compiere n'offre aucune documentation libre de développement, Openbravo maintient lui un wiki dédié tant aux utilisateurs qu'aux développeurs et qui s'étoffe de jour en jour. De même, le forum d'Openbravo est 10 fois plus fréquenté que celui de Compiere selon les chiffres de Sourceforge.net. Des gens d'Openbravo prennent également le temps d'y répondre et d'y diffuser tout leur professionnalisme. De plus, avec Openbravo, les développements ne seront ni propriétaires, ni l'apanage d'un l'éditeur hypertrophié (contrairement à l'interface web de Compiere).

Enfin, Openbravo poursuit une politique dynamique de recrutement de partenaires intégrateurs là où Compiere semble se satisfaire des monopoles de fait d'Axilom et d'Audaxis en France.

Les limites d'ERP5

Comme vous avez pu le constater dans ce livre blanc, ERP5 possède un périmètre fonctionnel très riche, sans doute le plus riche après TinyERP. Cependant, nous avons trouvé des limites importantes à ERP5.

Le handicap de la base Zope

Comme évoqué plus haut, ERP5 n'utilise pas de base relationnelle traditionnelle pour sa persistance mais utilise la 'base objet' Zope.

Or nous estimons que cette base Zope n'est ni suffisamment connue, ni suffisamment éprouvée pour qu'une entreprise décide d'y stocker ses données critiques à cette heure. De plus, la promesse de simplicité et de productivité de développement accrue liée à l'usage de cette technologie

ERP open source

est en partie annulée par le besoin d'une base relationnelle d'indexation, en redondance, pour assurer des performances décentes.

Ajoutons à cela que l'usage de Python comme langage de programmation n'aide pas. Non pas que celui-ci soit mauvais, bien au contraire – Google l'utilise par exemple à foison au même titre que Java – mais bel et bien parce qu'il s'agit d'un langage peu maîtrisé en environnement d'entreprise en France.

TinyERP est lui aussi écrit en Python et pour lui aussi ceci freine souvent son adoption. Mais les experts techniques remarqueront qu'à part à utiliser les tous récents langages dynamiques implémentés au dessus de la machine virtuelle Java comme JRuby, Jython ou Groovy, ni TinyERP, ni ERP5 n'auraient eu accès à une telle productivité. En effet, la langage Java a des limites intrinsèques qui cadrent mal avec les ERP: la définition des classes à la volée y est possible mais leur manipulation par l'API de réflexion reste l'apanage des codeurs experts et n'est pas du tout aussi lisible que le langage Java classique. Or un ERP moderne se doit justement de définir dynamiquement son modèle objet pour épouser les modélisations propres à chaque entreprise. Et il est souhaitable de dépasser les capacités standards introduites par le père de Compiere et son '*application dictionary*' repris dans les ERP qui en descendent.

En conclusion, **ERP5 accumule trop de spécificités techniques**. Dans le cas de TinyERP, parce qu'on reste sur une base relationnelle traditionnelle et parce que l'activité communautaire y est éclairante, l'usage de Python est rendu acceptable et ceux qui passeront la faible courbe d'apprentissage rentabiliseront rapidement leur effort. Nous estimons même que le retour sur investissement du développement s'y fera sentir plus rapidement que sur les ERP codés en Java, pourtant un langage balisé mais qui devra alors être balancé par la très grande complexité du framework.

ERP5: un business model où l'éditeur et l'intégrateur se confondent

Autre facteur de risque: l'unique intégrateur significatif d'ERP5 n'est autre que Nexedi, l'éditeur à l'origine d'ERP5. Tempérons néanmoins cette image de monopole par le fait qu'ERP5 reçoit quelques contributions de code, en provenance d'Amérique Latine par exemple. En ce sens, il reste un peu plus ouvert que Compiere mais on ne peut pas dire que la situation soit satisfaisante pour autant.

ERP open source

Voilà pourquoi ERP5, nous semble à réserver à:

- Un usage basique hébergé type ERP5 Express pour laquelle une TPE n'a ni besoin de comprendre comment marche son ERP, ni besoin de fonctionnalités non standard.
- Une intégration par Nexedi, éditeur d'ERP5; avec les risques que cela comporte en termes de dépendance d'un fournisseur unique.
- Une intégration personnalisée par les entreprises ayant en leur sein des experts Zope et qui n'ont pas peur d'en dépendre.

Récapitulatif sur Openbravo



Avantages

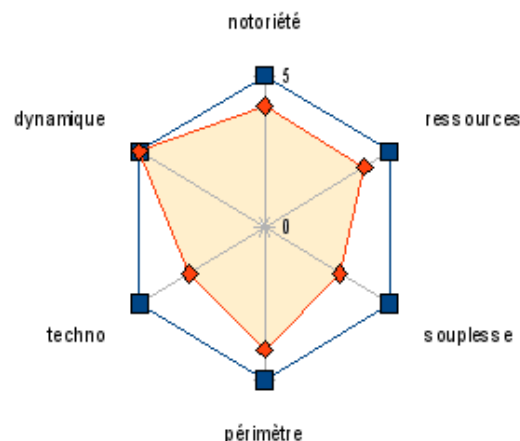
- **Éditeur très dynamique** qui investit beaucoup pour créer un puissant élan communautaire et marketing
- **Communauté** très dynamique en retour
- Éditeur solide apte à fournir un **support professionnel** digne de Compiere ou des ERP propriétaires
- Interface **web sans concession**, de loin la plus séduisante et la plus ergonomique
- Le domaine fonctionnel et **la fiabilité de Compiere**
- la tenue en charge de l'ERP n'évoque pas de crainte particulière. Il s'agit d'une webapp légère (pas d'EJB) classique sur un serveur Tomcat à laquelle on peut éventuellement adjoindre une base de données Oracle en cas de besoin de performance élevé.

ERP open source

Défauts

- Tout comme sur Compiere, la logique métier est codée à un faible niveau d'abstraction, ceci sera un fardeau pour l'évolution du produit à court terme, bien qu'à moyen terme une migration progressive vers une meilleure modélisation soit prévue.
- La souplesse dans les processus est souvent au prix de développement spécifique (pas de moteur BPM), bien qu'à moyen terme un moteur BPM BPEL soit envisagé.
- La francisation du module de comptabilité est à finaliser et ne pourra pas être considérée fiable courant 2008, un pont comptable sera donc requis dans un premier temps.
- Tout le modèle des données n'est pas exposé par webservice par la plateforme, il faut le faire au cas par cas

Profil général



Notoriété actuelle

On peut légitimement dire qu'Openbravo est sensiblement **aussi solide que Compiere** et qu'en ce sens on devrait au moins lui accorder la même confiance. Openbravo a par ailleurs connu un assez grand nombre de déploiements en Espagne ainsi que dans quelques autres pays européens. Citons par exemple, le distributeur de produits surgelés Frilac, l'industriel Huesker et bien d'autres références qui souhaitent pourtant ne pas être dérangées par des appels intempestifs. Pour autant, début 2008, il n'y avait toujours aucun déploiement en France. Ceci doit inciter à rester prudent vis à vis de la francisation de sa comptabilité générale.

ERP open source

Dynamique

De loin Openbravo est l'ERP qui possède **la plus forte dynamique** de croissance et ce, à une échelle mondiale. Ceci ce confirme tant en terme de fréquentation des forums, de communiqués de presse institutionnels ou de téléchargements. Ainsi Openbravo voit environ **1000 messages par mois** sur ses forums là où Compiere n'en voit plus que 100. Cependant, il faut relativiser cette dynamique car elle est aussi portée par un investissement important tourné vers ce business model, là où d'autres éditeurs sont auto-financés (ERP5, TinyERP), ou d'autres encore ont choisi un business model proche du modèle propriétaire (Compiere). Cependant, à l'instar d'Alfresco, cette politique volontariste de conquête peut faire mouche, attirer les talents qui démultiplieront la qualité et les fonctionnalités du produit. Le fait qu'Openbravo vienne à nouveau de lever **12 Millions de dollars** est sans doute l'une des illustrations de cette démarche volontariste.

Techno

Openbravo a sensiblement la même technologie de base que celle de Compiere, c'est-à-dire éprouvée, mais moins orientée vers l'agilité et le prototypage.

Le fonctionnel existant atteint déjà une masse honorable et a été fiabilisé par de nombreux experts de l'ERP. Ainsi Openbravo devrait nativement satisfaire à de nombreux cahiers des charges. D'ailleurs, de très nombreux ERP propriétaires réputés utilisent ces mêmes technologies, à l'instar de SAP R/3 par exemple avant que Netweaver et le module de BPM ne soient mis en place.

Sur Openbravo, les extensions fonctionnelles ne posent pas de problème particulier et sont conformes à celles qu'on est en droit d'espérer des développements classiques en J2EE, dans la mesure où la porte reste ouverte à tout type de choix de framework. Ainsi, rien n'empêche d'adjoindre à Openbravo une servlet Struts ou encore l'ORM Hibernate avant que l'éditeur ne le fasse lui même.

De plus, grâce à son volontarisme, l'éditeur est en passe d'atteindre une masse critique significative qui permet d'améliorer rapidement son produit. Ainsi, à moyen terme, Openbravo a un plan d'action crédible pour améliorer sa technologie: non seulement sa nouvelle plateforme dénommée 'Green' est techniquement à la pointe du développement Java (Spring, OSGI, Hibernate) mais Openbravo a des architectes de premier ordre pour l'implémenter. La migration progressive vers cette nouvelle plateforme 'Green' ne sera cependant pas effective avant 2009 et nous évaluons ici Openbravo tel qu'il est actuellement.

Périmètre

Le périmètre est sensiblement le même que Compiere, c'est-à-dire très adapté aux secteurs de la distribution et de la logistique. Cependant,

ERP open source

Openbravo va plus loin en terme de gestion de production et est ainsi également très indiqué pour des déploiements en industrie. En revanche, pour le secteur tertiaire, la gestion de projet est correcte mais assez minimaliste au regard d'un TinyERP.

ILLUSTRATION 24: APTITUDES PAR FONCTIONNALITÉ

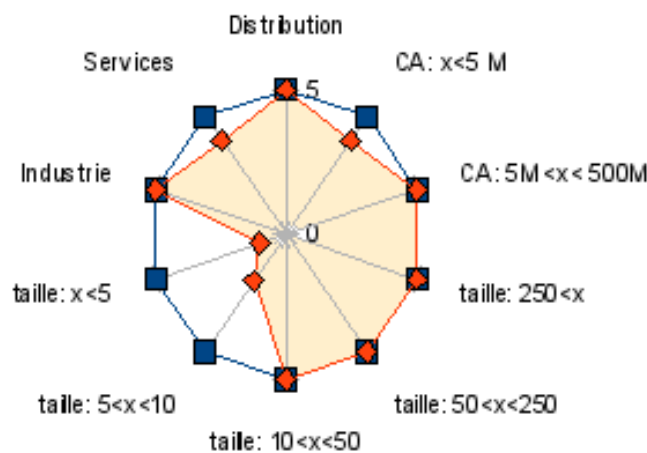
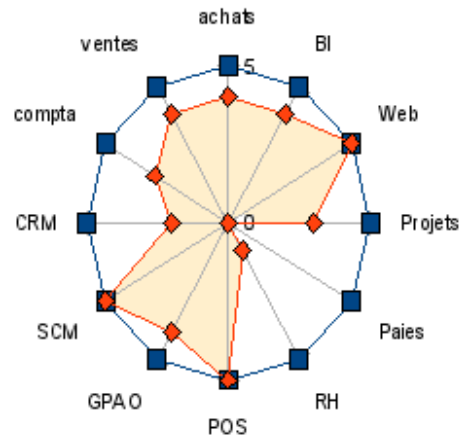


ILLUSTRATION 25: ADÉQUATION PAR RAPPORT AU SECTEUR ET À LA TAILLE DE L'ENTREPRISE

Souplesse

La souplesse est digne de Compiere pour ce qui est de la personnalisation des structures de données, de leurs écrans d'édition et des rapports. En revanche, comme sur Compiere, la souplesse d'intégration de nouvelles règles métiers ou d'altération des règles natives reste moyenne à cause de la lourdeur des cycles de développements et du niveau d'abstraction moyen seulement fourni par la plateforme actuelle (pas d'objets métiers). Le manque actuel d'un vrai moteur de workflow pourra aussi alourdir l'implémentation de certains process d'entreprise.

ERP open source

Web

Sans conteste la meilleure interface web! Un design très professionnel, une organisation intuitive et des raccourcis claviers efficaces.

Compta

Très bonne sur la partie analytique avec une foule de rapports paramétrables inclus et correspondant à des besoins réels. En revanche, la comptabilité générale devra elle se perfectionner au gré des déploiements hexagonaux.

BI

La Business Intelligence est pour l'instant limitée à du reporting. Mais celui-ci est très efficace avec de nombreux rapports d'analyses adaptés aux besoins réels et créés par des professionnels de l'ERP. De plus, un requêteur standard et compétitif de cube OLAP tel que Pentaho devrait être intégré à l'ERP d'ici l'été 2008. Dès lors Openbravo disposera d'une plateforme décisionnelle très avancée.

Ressources

On trouve facilement des ressources pour personnaliser ou maintenir Openbravo, d'autant plus que la documentation de l'éditeur est bonne et s'étoffe sensiblement de jours en jours. N'importe quel développeur Java expérimenté fera l'affaire. La note n'est néanmoins pas parfaite car le code est compliqué, avec plus de 450 tables et les abstractions mises en jeu gagneraient à s'élever plus haut au dessus de données relationnelles.

Techniques d'extensions fonctionnelles

Ajouter de nouvelles fonctionnalités à Openbravo ne pose pas de gros problème. En premier lieu, on peut étendre le modèle relationnel pour stocker de nouvelles données. Grâce à l'« *application dictionary* » hérité de Compiere, ceci peut être fait entièrement graphiquement, y compris l'adjonction de fenêtres pour l'édition type créer, lire, mettre à jour et supprimer, sur ces nouvelles tables. De même, la création de nouveaux rapports paramétrables est assez immédiate grâce à l'outil iReport.

Pour ce qui est des traitements sur ces données, plusieurs solutions existent: les adeptes du pl/SQL pourront très facilement ajouter un bouton qui déclenchera une procédure stockée qu'il auront écrite. On pourra préférer un appel vers une servlet Java. En revanche, Openbravo ne mettra pas ici à notre disposition d'abstraction très puissante et la couche d'accès aux données par le framework appelé SQLC ne permettra que de gérer les opérations simples telles que créer, lire, mettre à jour et supprimer avec la base de données. Les notions d'associations de tables ou d'héritage devront être traitées au cas par cas.

ERP open source

Bien sûr, rien n'interdit de mettre en place des outils plus puissants tel qu'Hibernate pour manipuler ces données, mais avant que la plateforme Green ne soit opérationnelle courant 2009, il faudra en assumer le coût d'infrastructure.

On touche donc là aux limites de l'implémentation actuelle d'Openbravo: à cause de ces faibles degrés d'abstraction, il est assez coûteux de modifier les fonctionnalités natives de l'ERP, comme par exemple les règles de pricing ou de sélection d'un produit dans un stock. En effet, le manque d'encapsulation du code essentiellement SQL demandera toujours de grandes précautions pour éviter les effets de bords indésirables.

Conclusion

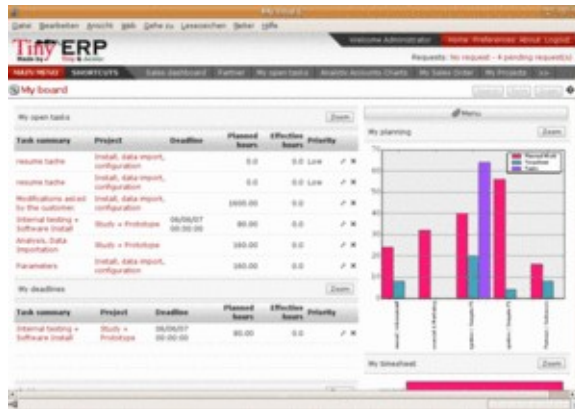
Oui, il existe bel et bien un ERP qui a la fiabilité d'un Compiere, la force d'un éditeur très professionnel apte à garantir un support de qualité, mais qui reste ouvert à la fois à la mise en concurrence des intégrateurs et aussi à la documentation de son développement. Il s'agit d'Openbravo.

A l'heure actuelle, bien que son interface web tranche nettement, Openbravo ressemble encore à Compiere en terme de fonctionnalités mais le dépasse déjà pour ce qui est de la gestion de production industrielle ou de gestion des points de vente. A l'avenir, il est probable que cette démarche d'ouverture paie et se traduise par des extensions fonctionnelles qui seraient bienvenues, comme la gestion des ressources humaines par exemple.

De même, la technologie utilisée reste encore proche de celle de Compiere, c'est-à-dire que les écarts au fonctionnel existants se paient lourdement (au delà de la simple altération du modèle de données et des écrans donc) mais cela devrait rapidement s'améliorer avec l'intégration progressive de la nouvelle plate-forme Green qui permettra des développements plus productifs et plus de symbiose entre les divers développements tierces qui ont déjà lieu un peu partout dans le monde.

Enfin, la francisation qui demande encore de la finition et le besoin de développer les quelques fonctionnalités qui pourraient manquer à des besoins spécifiques rend Openbravo encore peu adapté pour les TPE françaises. En revanche les PME d'une certaine taille, qui peuvent rentabiliser les investissements requis, seront rassurées par les capacités d'Openbravo en termes de tenue de charge.

Récapitulatif sur TinyERP (OpenERP)



Avantages

- **Éditeur très dynamique**
- Communauté dynamique et expérimentée
- Périmètre fonctionnel inégalé avec ses quelques 200 modules et des nouveaux modules tous les mois.
- **Conception très intelligente.** Souvent jusqu'à 10 fois moins de code que les ERP en Java pour offrir les même fonctionnalités!
- Interface **web** très compétitive
- Vrai ORM qui fait le pont entre la base relationnelle et le code objet proche des spécifications fonctionnelles
- Tout **le datamodel et les méthodes métier sont nativement exposés en webservices**, c'est un gage d'interopérabilité facile
- Moteur **BPM intégré** très efficace
- Grand souplesse générale, notamment grâce à la scriptabilité des rapports
- Croissance auto-financée

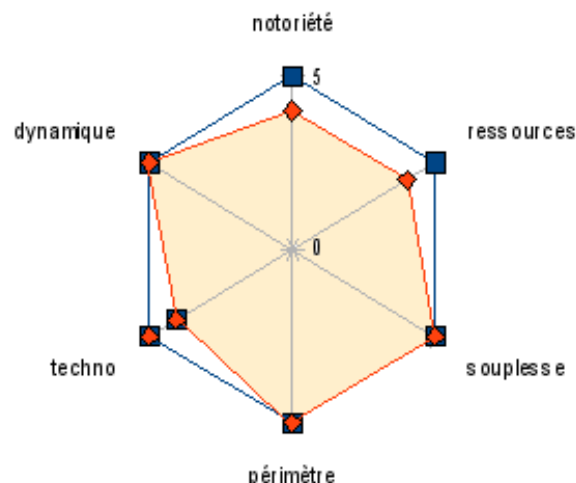
Défauts

- Python n'est pas toujours très accepté en entreprise, du moins en France
- Manque de professionnalisme du marketing, ceci freine à la croissance de la communauté, professionnelle notamment.

ERP open source

- l'ORM, bien que puissant n'est pas un standard reconnu
- bien que redoutablement efficace, le moteur de BPM aurait gagné à être un standard reconnu. Tempérons néanmoins cette critique car une compatibilité XPDL peut être assurée.
- pour un grand compte ou un très grand nombre d'utilisateurs simultanés (> 100), **la tenue en charge reste à démontrer**. Cela pourrait marcher, mais il n'y a pas encore de retour d'expérience pour le prouver.
- Les webservice sont XML/RPC et pas SOAP. Ceci est donc un léger handicap pour l'interopérabilité avec des outils graphiques qui introspectent les descripteurs WSDL de webservice SOAP.

Profil général



Notoriété actuelle

Plusieurs dizaines voire centaines de déploiements dans le monde entier, de l'Argentine à la Chine en passant par l'Inde. Mais encore assez peu de grosses PME telles que celles qui ont fait la renommée de Compiere. Citons pourtant parmi les références les Hotels de luxe Costes (Sednacom), Whirlpool Paris, l'administration du canton de Vaud (Suisse), IR-Microsystemps... Étant donné le potentiel du produit, de nouvelles références importantes ne devraient pourtant pas tarder.

Dynamique

La dynamique est aussi très forte. La société éditrice est passée de moins de 5 à plus de 60 salariés en moins d'un an et demi pour répondre à une demande en très forte croissance. De même, le nombre d'intégrateurs s'étoffe significativement de mois en mois dans le monde entier.

ERP open source

Techno

Sans doute l'ERP open source le plus moderne au plan technique. La souplesse de modélisation d'un ERP5 mais la base relationnelle d'un Compiere.

Pour autant, on pourra regretter que ni l'ORM ni le moteur de BPM ne soient des standards reconnus. De même, si l'usage d'un langage dynamique tel que Python pour les couches métiers de l'ERP participe indubitablement à la souplesse inégalée de l'outil, pour les couches basses d'infrastructure, un langage statique tel que Java aurait apporté un gain de performance et de fiabilité. Notons cependant que cette fiabilité semble pourtant assurée dans le cas de TinyERP par une large batterie de tests unitaires et une très large communauté d'utilisateurs et de développeurs vigilants. On aurait donc préféré du Jython/JRuby ou Groovy+Java que du Python pur. Mais comme l'ERP idéal n'existe pas et étant données les contraintes existantes lors de sa création, TinyERP mérite déjà largement le meilleur classement en terme de technologie.

Périmètre

Là aussi, le plus vaste périmètre fonctionnel grâce à ses quelques 200 modules. Si 50% de ces modules relèvent d'un certain amateurisme, il en reste néanmoins une large base de modules réellement efficaces. Outre les domaines classiques, il y a une foule de modules variés dédiés à des cas très spécifiques: tels que la création de portails pour les clients, la gestion des adhésions aux associations, la gestion de projet informatique agile (SCRUM)...

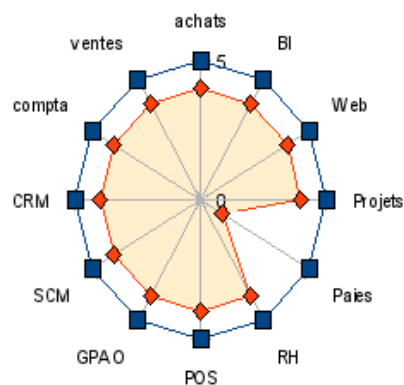


ILLUSTRATION 26: APTITUDES PAR FONCTIONNALITÉS

ERP open source

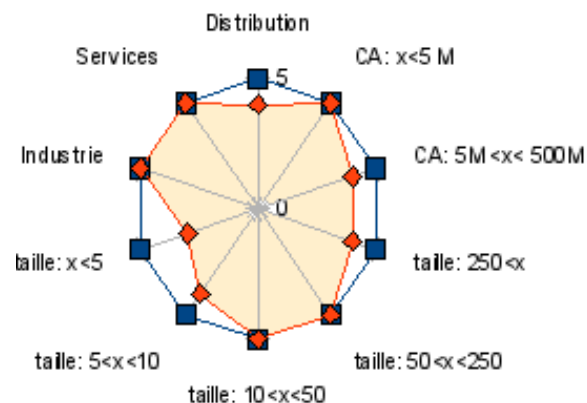


ILLUSTRATION 27: ADÉQUATION PAR RAPPORT AU SECTEUR ET À LA TAILLE DE L'ENTREPRISE

Souplesse

Très bonne souplesse grâce à la scriptabilité de bout en bout et plus spécifiquement dans les workflows et le reporting. Par ailleurs, le puissant moteur de workflow mis en œuvre par TinyERP est une des clés de sa souplesse.

Web

eTiny, la surcouche serveur développée initialement par Axelor, puis désormais co-développée par Axelor et Tiny.be est un modèle de simplicité et d'efficacité. Elle ne fait que traduire les webservices de TinyERP en HTML et apporte des fonctionnalités avancées comme l'autocomplétion Ajax ou les raccourcis clavier. La couche web ajoute même un composant qui permet de visualiser les plannings, il s'agit de la seule différence sensible avec le client lourd.

Compta

Comme sur d'autres ERP, la comptabilité analytique est compétitive: gestion des budgets, comptabilité analytique multi-axiale et hiérarchique. Concernant la comptabilité générale, bien que l'une des plus avancée et bien qu'utilisée dans certaines TPE, elle reste toujours à perfectionner pour atteindre les standards exigés par les comptables. Notons toutefois qu'en dépit de ces manques de finitions, le plan comptable français est inclus que Tiny permet l'édition des bilans, comptes de résultats et liasses fiscales. Au final, nous estimons qu'on se dirige pour 2009 vers une comptabilité générale sans problème (au même titre qu'une ligne Sage 100 par exemple) mais que pour l'instant, les manques de finitions il est préférable d'utiliser un pont comptable (cf page 50) ou de transmettre les extractions à un expert comptable, à moins que l'entreprise ne soit en mesure à la fois de dominer les règles comptables et aussi d'appliquer des correctifs mineurs à l'ERP en cas de bug.

ERP open source

BI

La Business Intelligence se limite encore à des rapports paramétrables. Ceci dit, TinyERP inclura également une solution de requêteur de cube OLAP dès l'été 2008 pour des analyses plus fines et sans coût d'intégration démesuré.

Ressources

Trouver des gens qui savent coder en Python est plus répandu qu'on ne le croit mais pas aussi facile que trouver un développeur Java ou PHP, certes. Cependant, il vaut mieux parfois apprendre les rudiments d'un nouveau langage plutôt que de rester dans un langage classique et affronter en contrepartie des frameworks complexes et moins productifs.

Enfin, contrairement aux idées reçues, Python dispose aussi désormais d'un environnement de développement gratuit et compétitif. Par exemple PyDev sur Eclipse, dispose d'un débogueur et de fonctions de navigations dans le code avancées, similaires à ce qui existe pour Java. Enfin, le recours à du code SQL classique abondamment utilisé sur les ERP de la lignée Compiere, continuera bien sûr de marcher sur la base PostgreSQL si les outils de haut niveau ne conviennent pas. De même, l'ouverture totale par webservices simples XML/RPC rend enfantin l'intégration de TinyERP avec d'autres progiciels, quelle que soit leur technologie.

Techniques d'extensions fonctionnelles

Tout comme sur Openbravo, la création de nouvelles structures de données, de nouvelles interfaces d'éditions et de nouveaux rapport est assez immédiate et peut se faire par des interfaces graphiques.

Pour ce qui est des traitements sur les nouvelles structures de données, la méthode préférée est l'écriture de code Python, simple, manipulant les objets correspondants à ces données par l'intermédiaire de la couche de mapping relationnel. Ce framework prendra soin de gérer pour nous les associations entre tables ainsi que les héritages, ce qui amène à un code très lisible, au plus poche des spécifications fonctionnelles. En revanche, si on ne désire pas apprendre les subtilités de ce framework, on pourra, sur TinyERP aussi, déclencher du code SQL, éventuellement sous forme de procédures stockées.

De plus, l'une des forces de TinyERP, c'est de fournir simplement tous les outils permettant à chacun de re-packager ses développement spécifiques en un module indépendant et réutilisable: soit dans une simple optique de maintenabilité, soit dans une optique de construction communautaire (cf page 72).

Enfin, n'oublions pas que TinyERP embarque nativement un moteur de BPM puissant qui permettra, toujours par le biais d'interfaces graphiques de définir de nouvelles règles de gestion à plusieurs étapes faisant intervenir différents rôles fonctionnels là où il faudrait au contraire se lancer dans un code hasardeux avec beaucoup d'autres ERP.

ERP open source

Conclusion

Une analyse rationnelle de TinyERP devrait vous amener à réviser votre jugement sur Python: il s'agit de l'ERP le plus riche fonctionnellement, il est véritablement ouvert et très facile à personnaliser pour satisfaire à des besoins très spécifiques tout en maintenant une base de code propre. De la MRP à la CRM avancée en passant par la gestion de projet avec des tâches individualisées, cet ERP excelle dans tous les domaines. En fait, la richesse fonctionnelle n'est qu'une conséquence à la fois de la démarche d'ouverture et de la très bonne technologie qui rend les développements tiers simples et nombreux: dès lors presque toutes les intégrations donnent lieu à un feedback positif pour toute la communauté sous forme de modules proprement packagés. Seul la tenue en charge devra encore faire ses preuves dans des entreprises de plus en plus grosses. Pour les grosses entreprises, cela veut dire que pour l'instant il faudra prévoir de conduire des tests de charge. A terme nous ne voyons aucune limite à l'expansion de cet ERP.

Récapitulatif sur Neogia



Avantages

- **Scalabilité avérée** de la solution
- Conception correcte, **principes modernes**
- e-commerce intégré et extensible (scalable) pour le B2C, notamment pour des applications d'e-commerce à forte charge.
- Open source au sens le plus stricte du terme (communautaire)

ERP open source

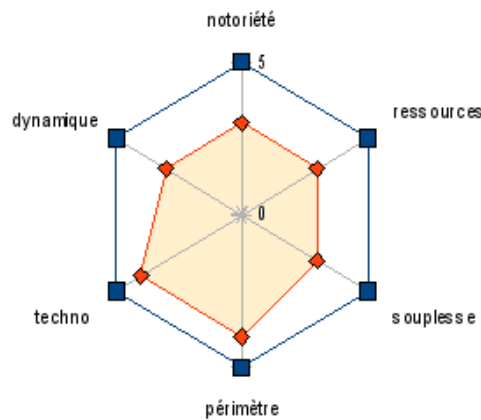
- Indépendance vis à vis de la base de données (pas de procédure stockée)
- Orientée service: tout le modèle de données est exposé en **webservices**
- Comptabilité analytique multi-axe
- Prototype d'intégration du serveur de BI Pentaho

Défauts

- Le développement de fonctionnalités spécifiques additionnelles est au prix d'un investissement initial relativement lourd. En effet, la modification du modèle de données et des process passe systématiquement par des phase de développement comprenant génération de code, compilation puis enfin de déploiement du code en production.
- Courbe d'apprentissage plate, donc coût **minimal d'implémentation assez élevé**.
- La comptabilité française doit être abordée avec prudence
- Si Neogia est techniquement plus satisfaisant qu'Ofbiz, sa pérennité n'est pas garantie selon nous car la communauté n'a pas la masse critique requise et par ailleurs **il n'y a pas non plus d'éditeur fort pouvant investir** dans le projet. En effet, en France, Nereide est le seul intégrateur de taille significative et cette situation commence à durer. OfBiz étant pérenne car porté par Apache et fort d'une communauté internationale importante, nous pensons néanmoins qu'en cas de déliquescence de Neogia, le système installé ne serait pas en souffrance car il reste très largement basé sur OfBiz
- La modélisation par UML reposait sur l'outil Poséidon UML qui est devenu payant. Neogia affronte donc une période transitoire pendant laquelle le développement productif de fonctionnalités spécifiques n'est plus assuré par des outils libres. Lorsqu'on connaît la complexité des outils générateurs de code à partir d'UML, on se demande si l'idée même est réellement pérenne. La programmation déclarative des relations entre objets à la manière de Rails paraît plus viable et même plus simple que la dépendance à de tels outils visant à rattraper les faiblesses des langages statiques.
- l'interface utilisateur est minimaliste et, il faut bien l'admettre, assez laide (comparée à celle d'Openbravo par exemple) et pourrait éventuellement donner lieu à un rechartage coûteux.

ERP open source

Profil général



Notoriété actuelle

Si la notoriété d'Open For Business (OfBiz) n'est plus à faire à l'international, dans l'hexagone, la notoriété du dérivé qu'est Neogia reste à conquérir. En France, si on trouve quelques entreprises ayant choisi OfBiz pour leur plateforme e-commerce, à l'instar d'Yves Rocher, peu d'entreprises utilisent encore Neogia comme véritable ERP. On trouve cependant les industriels Transrail Boige & Vignal, Saitek ou Messer Group, tous intégrés par la société Nereide.

Dynamique

Bien qu'Ofbiz semble continuer à croître à l'échelle mondiale, concernant Neogia des doutes sont permis. En effet, en dépit d'une activité perceptible sur 2007, le site de l'éditeur n'a pas fait beaucoup de bruit depuis 2008 et la situation contraste nettement avec Openbravo, Compiere ou TinyERP. Mais répétons le, comme la base de code est propre et le projet ne fait que dériver d'Ofbiz, la situation pourra toujours évoluer favorablement.

Techno

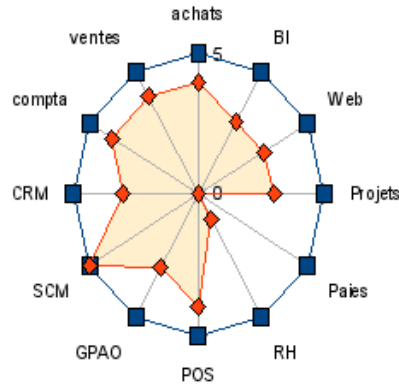
C'est le point fort de Neogia: sa technologie est meilleure que celle des ERP de la lignée Compiere et presque aussi bonne que celle de TinyERP, tandis que sa plateforme Java est connue et maîtrisée en entreprise.

En effet, tout comme OfBiz, Neogia dispose d'un ORM assez avancé (mais pas standard pour autant hélas) qui, outre la persistance, gère aussi le modèle relationnel. Dans OfBiz, cet ORM ne va cependant pas aussi loin que TinyERP ou ERP5 dans la mesure où les objets métiers n'ont toujours pas de méthodes propres suivant le modèle objet. Ceci change avec Neogia et son générateur de modèle qui permet aux objets métiers d'acquérir une spécialisation de comportement, à l'instar de ce qui est fait dans ERP5 ou TinyERP.

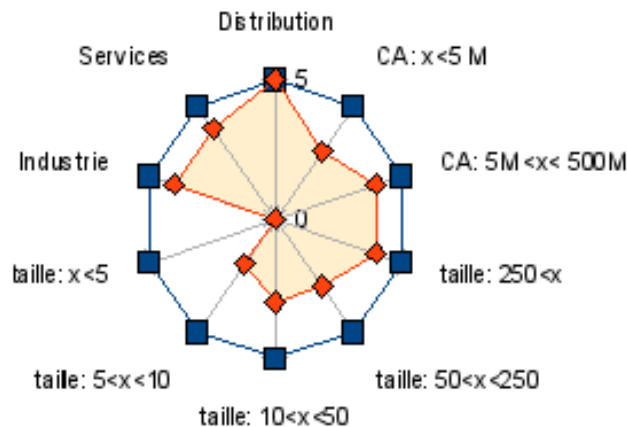
ERP open source

Périmètre

Le périmètre de Neogia est très correcte mais manque souvent de finition, comme bien souvent avec les solutions essentiellement communautaires.



28: APTITUDES PAR FONCTIONNALITÉS



29: ADÉQUATION PAR RAPPORT AU SECTEUR ET À LA TAILLE DE L'ENTREPRISE

Souplesse

La souplesse de Neogia est bonne, meilleure que celle des ERP de la lignée Compiere grâce à sa modélisation objet plus poussée.

Web

Neogia, tout comme OfBiz est d'abord une solution web. Pour autant l'interface manque cruellement de finition et il est possible que les utilisateurs vous le reprochent. Tempérons néanmoins car un ERP comme SAP pêche lui aussi par une interface encore moins conviviale.

ERP open source

Compta

Les fonctionnalités de comptabilité analytique sont très avancées. En revanche, pour ce qui concerne la comptabilité générale, en dépit des annonces de l'éditeur qui y a travaillé d'arrache pied, il conviendra de rester prudent et de passer par un pont comptable en premier lieu. En effet, le faible nombre de déploiements et la faible activité de la communauté française permet difficilement d'utiliser ce module sereinement en l'état.

BI

Comme sur les autres ERP, les fonctionnalités avancées de reporting sont au rendez vous et sont fournies cette fois par l'excellent outil BIRT. De plus, un 'proof of concept' d'intégration avec le serveur de requête de cube OLAP Pentaho a été mis en place par Nereide et devrait être facilement adaptable à d'autres cas avec un peu de travail.

Ressources

Des programmeurs Java expérimentés sauront implémenter de nouvelles fonctionnalités dans Neogia, mais en dépit d'une solution assez bien architecturée, ceci sera rendu plus difficile par la masse critique encore trop faibles des intégrations actuelles.

Techniques d'extensions fonctionnelles

Sur Neogia, altérer les structures de données nécessite de modifier des fichiers XML avant de générer du code puis de recompiler et de redéployer l'ERP. Si les abstractions sont en place, ceci ralentit néanmoins considérablement le prototypage des solutions par des consultants fonctionnels. Notons aussi, que la couche de mapping relationnel qui fait le lien entre une certaine modélisation objet et les tables en base de données gère, elle, au moins les associations, et même les notions d'héritage pour ce qui est de Neogia (à la différence d'Ofbiz). Pour les traitements sur les nouvelles structures de données, le code Java qui utilise ce framework est à privilégier. En conséquence, le code fonctionnel existant sur Neogia, est lisible et maintenable. En ce sens, à défaut d'être nativement très compétitif, cet ERP s'appuie sur une socle sain apte à être étendu dès lors qu'on en assume les investissements.

Conclusion

Neogia est une **solution saine** en tout point mais qui hélas ne décolle toujours pas et, en conséquence, manque de finition. Dans ces conditions il paraît bien difficile de passer par un autre intégrateur que l'historique Nereide qui maintient le projet. Ne faisons néanmoins aucun mauvais procès à cet éditeur/intégrateur puisqu'il agit en totale ouverture en documentant très largement sa solution, en espérant que d'autres intégrateurs se joignent enfin à son effort. De plus, à la différence de

ERP open source

Compiere ou d'ERP5, Nereide ne détient pas un gros monopole sur sa solution puisque celle-ci hérite à 80% d'Ofbiz qui est une solution ouverte et populaire à la fois.

Du fait des manques de finitions et justement de l'absence d'éditeur fort, Neogia est **difficilement un ERP clé en main**. Il s'agit plutôt d'un ERP qui nécessitera un investissement initial assez conséquent pour être intégré. En ce sens, mis à part le cas de TPE comportant exceptionnellement de fortes compétences Java, Neogia s'adresse difficilement aux entreprises de moins de 40 personnes. Et au delà de cette taille, une entreprise pourrait être au contraire plus rassurée par un éditeur fort apte à fournir un support professionnel de qualité tels que ceux qu'on trouve derrière Openbravo, Compiere ou ERP5. Au final la situation n'est pas si simple pour Neogia mais une voie reste ouverte si votre philosophie est vraiment d'**investir dans une solution avant tout libre et indépendante** et Neogia sera particulièrement indiquée **si vous devez réaliser des développements spécifiques importants et que vous exigez une plateforme Java ou encore en cas de besoin de plateforme d'e-commerce très robuste**.

SYNTHÈSE

Comme nous l'avons vu les ERP open source sont en plein essor et on devrait enfin assister à leur arrivée à maturité dans tous les domaines fonctionnels des ERP génériques alors que jusqu'ici seuls des précurseurs comme Compiere se cantonnaient à des domaines fonctionnels très spécifiques comme la distribution et la logistique.

En France, **Openbravo, TinyERP, Neogia, Compiere, Adempiere** et **ERP5** sont incontestablement les acteurs de cette mutation.

Le dernier rempart à une plus large adoption de ces ERP open source est leur manque de maturité sur la comptabilité générale française. A cause de ceci, il faudra soit assumer un temps de paramétrage important du module de comptabilité, soit contourner l'obstacle en mettant en place un pont comptable vers une solution du marché ou bien fournir ces données comptables à votre expert comptable s'il est indépendant.

Sur le plan technique, il est difficile de concilier une persistance des données dans une base relationnelle avec l'impératif de souplesse vis à vis de modèles de données susceptibles de changer d'une entreprise à l'autre, ceci avec une modélisation unifiée de haut niveau.

Ce compromis n'a pas été négocié de façon convaincante ni par ERP5 qui accède certes à cette modélisation fonctionnelle de haut niveau mais au prix d'un choix technique exotique, la base objet Zope, ni par la lignée des Compiere qui restent trop près des données SQL.

Outre ces aspects fonctionnels et techniques, dans l'open source il ne faut pas négliger l'impact des business models et de la gouvernance. Or, force est ici de constater qu'encore une fois ERP5 et Compiere semblent essayer de se reposer sur leur relative notoriété et ne soutiennent pas un modèle de développement ouvert en dépit de leur licence open source. Or, du fait de la grande complexité d'un ERP, en l'absence de communauté d'utilisateurs et de développeurs active et de documentation, il faut relativiser l'aspect open source. En conséquence, il ne faudra pas s'étonner que peu d'intégrateurs se bousculent pour les intégrer et ces situations de quasi monopoles ne favorisent ni l'accroissement du périmètre fonctionnel, ni la baisse des coûts d'intégration.

Au contraire, selon nous TinyERP, Openbravo et Neogia se distinguent à la fois par des démarches d'ouverture assumées et par une surenchère constante de fonctionnalités qui rivalisent dès aujourd'hui avec les solutions propriétaires à des coûts bien inférieurs.

Neogia, bien que basé sur une technologie à la fois assez bonne (abstraction objet indépendante de la base relationnelle) et connue (Java), est à réserver à des cas où ce n'est pas la compétitivité native qui prime sur le besoin d'implémenter beaucoup de fonctionnalités spécifiques. En

ERP open source

effet, Neogia manque quelque peu de finition et aucun éditeur n'est là pour assurer un support professionnel digne des ERP propriétaires sur cette solution foncièrement communautaire et dont la communauté française n'est pas très active. En revanche, étant basé sur OfBiz, Neogia disposera d'une base assez saine pour développer des fonctionnalités poussées tout en assurant la maintenabilité des développements. De même, de forts besoins en e-commerce sur une plateforme Java jouent en la faveur de cet ERP.

TinyERP (rebaptisé OpenERP) représente un idéal de logiciel agile, apte à répondre à n'importe quel besoin, tant qu'on n'est pas soumis à des charges critiques, du moins en attendant de plus amples retours d'expériences. TinyERP combine à la fois la force d'un éditeur et une réelle communauté qui balise la plupart des cas d'usages et fournit de précieux retours, notamment sous forme de modules réutilisables. Tout ceci est rendu possible par une réelle innovation technologique qui s'appuie néanmoins sur des standards reconnus et terme de base de données et de webservices.

Openbravo est un logiciel rodé sur son périmètre fonctionnel, certes un peu moins large, mais qui conviendra très souvent, notamment pour les secteurs de la distribution, la logistique ou la fabrication. Les fonctionnalités et la technologie actuelle se rapprochent de Compiere mais Openbravo va plus loin en terme de gestion de fabrication et propose un business model plus ouvert. Openbravo propose par ailleurs un cheminement progressif et convaincant pour élever son niveau de modélisation bien au dessus de Compiere. Sa dernière levée de fond de 12 millions de dollars rend ce plan d'action totalement crédible et serein. En l'état, on le privilégiera sur Neogia si on ne compte pas investir lourdement dans des développements spécifiques et si le périmètre natif convient.