

Y A-T-IL UNE PEDAGOGIE DE L'INFORMATIQUE ?

ENTRETIEN AVEC DEFI FORMATION RÉALISÉ PAR JACQUES VÉTOIS

Les entreprises de formation à l'informatique sont nombreuses sur le marché. Ce sont souvent de jeunes PME ouvertes et pragmatiques sur le plan pédagogique. Nous avons demandé à Gérard Louvet de DEFI Formation de nous expliquer son approche de la formation des utilisateurs de micro-informatique professionnelle.

Terminal: Pouvez-vous présenter DEFI Formation ?

Gérard Louvet : DEFI Formation a été créée en 1984 par une équipe de personnes qui avaient déjà une expérience de la formation permanente à l'informatique pour des publics de salariés. Certains possédaient une culture plutôt technique, d'autres plutôt organisationnelle. Aujourd'hui, ce sont douze personnes qui assurent essentiellement une activité de formation, de conseil et de recherche pédagogique. Nous avons deux axes d'interventions : les utilisateurs et les informaticiens. Pour nous, les utilisateurs ne sont pas des opérateurs passifs d'applications fermées, mais d'authentiques développeurs d'applications à l'échelle d'un service sur des progiciels et des langages micro ou infocentre. Pour les informaticiens, nous avons développé un cursus pédagogique allant des langages au génie logiciel, en nous appuyant sur la recherche publique et l'industrie.

T. Les formateurs ont surtout une activité de formation. Comment arrivent-ils à suivre les développements de la recherche dans leur domaine ?

G.L. Dans les formations de pointe, nous faisons intervenir des ingénieurs et des chercheurs extérieurs qui apportent leurs connaissances scientifiques et techniques. Par contre les con-

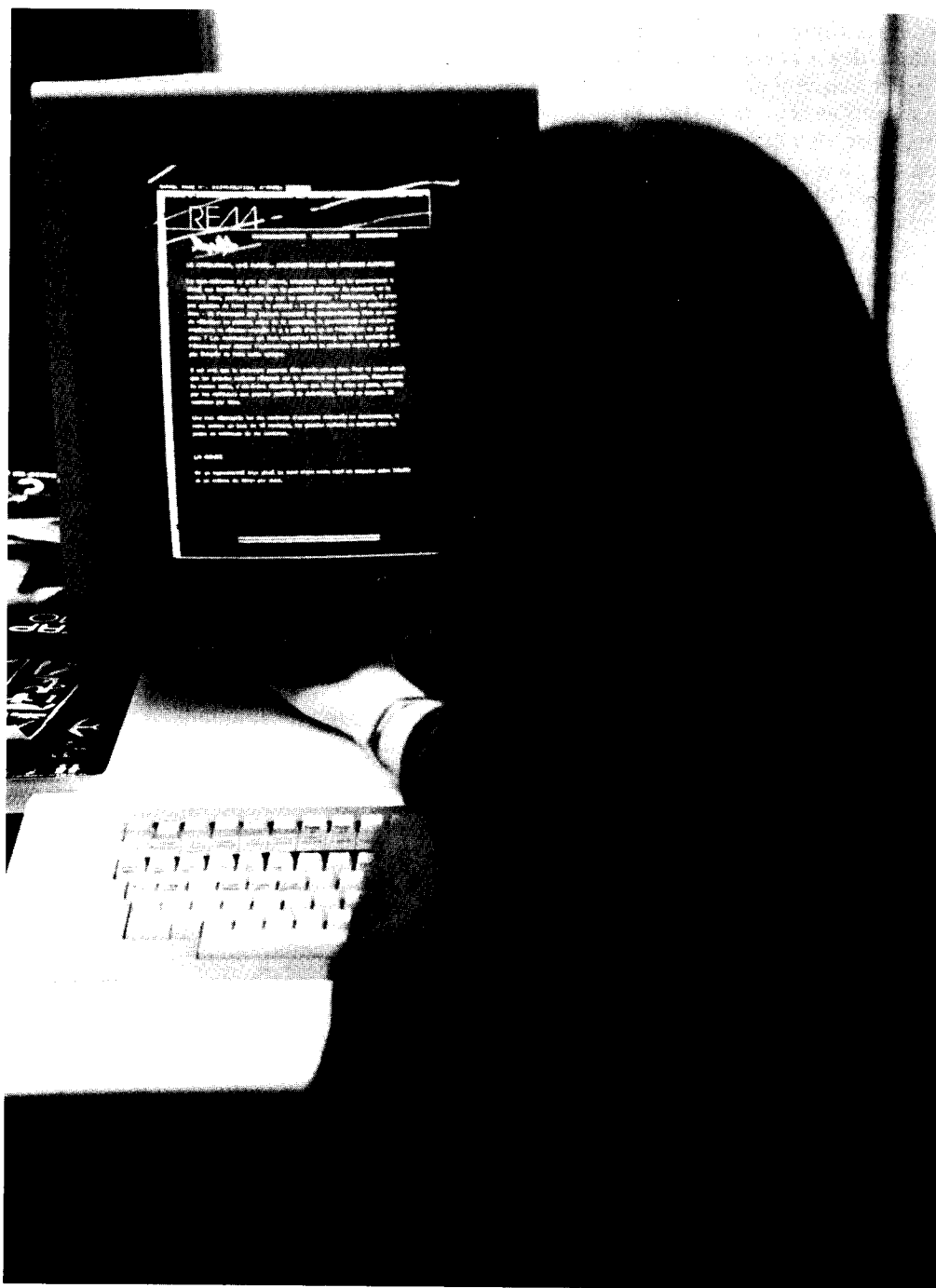
cepts de base, l'algorithmique, la programmation, la modélisation, etc., sont mieux enseignés par des pédagogues. A titre d'exemple, les formations d'ingénieurs logiciel de quatre mois sont réalisées à 50% par des formateurs internes et à 50% par des intervenants externes. La recherche pédagogique se fait en étroite collaboration entre les consultants de DEFI Formation et les intervenants extérieurs. C'est comme cela qu'une sorte d'osmose continue s'établit entre nous et nos partenaires. Certes, ils ont une pratique plus soutenue, mais en retour ils nous disent trouver à DEFI un lieu de synthèse et de structuration de leur pensée et de leur savoir-faire.

De toutes façons, tous les ingénieurs de DEFI assurent du conseil aux entreprises clientes, dans des proportions variables. Si l'on ajoute à cela les formations permanentes organisées en interne ou auxquelles l'un ou l'autre s'inscrit à l'extérieur, l'alimentation en compétences et en idées nouvelles ne manque pas. Le problème est plutôt parfois de ne pas nous disperser, tellement l'informatique évolue vite.

T. Pouvez-vous nous parler plus en détail des méthodes pédagogiques pour la formation de base des « utilisateurs » ?

G.L. Les concepts fondamentaux des progiciels outils sont toujours les mêmes. Dans chaque famille de progiciels, il existe un certain nombre de concepts et de fonctions semblables. Pour un tableur la notion de cellule, de fonctions, de copie de cellules, etc., pour une base de données, la notion de champ, de langage de requête, de mécanisme relationnel...

Mais plus fondamentalement que cela, il y a la modélisation d'une situation du monde réel. Qu'est-ce qu'on veut faire ? Quelles sont les données que l'on veut utiliser ? Quelles sont les relations entre elles ? Quand un utilisateur de progiciel bute sur une difficulté, ce n'est pas toujours un problème technique mais souvent parce qu'il n'a pas été capable de modéliser la réalité pour en faire ensuite des fichiers et des procédures. On s'attache donc beaucoup aux aspects sémantiques du système d'information pour en déduire les représentations les plus simples possibles. Pour cela, on peut utiliser des concepts développés par des profession-



nels, par exemple le modèle entité-relation. Nous en avons extrait des éléments essentiels pour nos formations « utilisateurs ».

On ne peut pas non plus concevoir une formation à Dbase III sans fournir au départ des éléments de structuration des traitements. Dbase est un langage procédural proche du Pascal couplé à une base de données. Mais il faut d'abord introduire les notions élémentaires d'algorithmique et pour cela, nous avons construit un petit logiciel de gestion commerciale. On demande à chaque groupe de trois sta-

giaires de dérouler l'application sur un micro. Le premier jeu consiste à représenter la succession des écrans et son architecture par des pastilles de différentes couleurs au tableau. En général, les stagiaires n'avaient jamais pris conscience de la structure arborescente sous-jacente à un dialogue interactif. Ce jeu n'est pas bloquant. Tout le monde y arrive. On en profite pour parler également de l'ergonomie des écrans, on envisage d'autres solutions. Ensuite, on veut faire comprendre que derrière un menu affiché à l'écran, il y a un traitement auto-

matique, donc un algorithme. Le groupe va noter le déroulement des questions et des réponses, comprendre les séquences et les traduire dans un langage simplifié (en français). Rapidement, on peut introduire des concepts de la programmation, puis on passe à la notion de fichier. On détaille les algorithmes les plus évidents : création, mise à jour et suppression.

Dans ces stages utilisateurs, on fait en sorte que les participants rencontrent les classes de problèmes les plus fréquents qu'ils seront amenés à automatiser dans leur bureau. C'est la pédagogie de la résolution de problèmes : comment s'y prendre dans telle situation (et non pas : voici toutes les commandes que le progiciel peut exécuter ; il peut le faire, mais pourquoi ?). C'est la pédagogie du 80/20 : 20% des commandes résolvent 80% des problèmes, ce sont celles-là qu'il faut impérativement enseigner et dont il faut apprendre comment les enchaîner correctement pour une gestion de trésorerie, un suivi budgétaire ou une relance clients.

Nos clients nous ont aussi poussé à développer des ateliers logiciels. Y viennent de faux débutants qui, comme beaucoup en anglais, ont un vocabulaire fourni, mais sont dans l'incapacité de construire deux phrases correctes.

T. Vous ne rencontrez jamais ce que j'appellerais des « bidouilleurs fous » ?

G.L. Si, mais rapidement, ils « se plantent ». Il faut savoir aussi utiliser la pédagogie de l'échec à condition de le rattraper. Il est courant d'entendre dire qu'un bon stage informatique c'est un stage où on manipule beaucoup sur machine. Il est vrai que certaines notions sur les tableurs s'expliquent bien plus facilement devant un écran. Mais notre méthode, c'est toujours le pourquoi avant le comment.

On analyse d'abord un problème au tableau par petits groupes, on décrit ce que l'on veut faire et, après seulement, on passe sur la machine. Les programmes étant bien conçus, ils sont mis au point très rapidement. C'est cette méthode que l'on applique par exemple dans un stage comme « *Conception d'applications en Open Access* ». A la fin du stage, les participants écrivent un programme de 1000 lignes, cinq ou six fichiers, des index, une base de données de gestion de stocks. Naturellement, quand nous formons des ingénieurs, nous allons plus vite à des phases d'exposés plus abstraits.

Quelqu'un qui ne possède comme bagage informatique qu'une initiation à Open Access se dit « Je n'y arriverai jamais. » Pourtant, après avoir analysé le problème par groupe de trois, défini les règles de gestion à appliquer, conçu la structure de la base de données, on s'aperçoit que la programmation se fait en deux jours. On démontre ainsi concrètement qu'il faut beaucoup concevoir et passer peu de temps devant l'ordinateur. De plus, cette démarche méthodologique est un investissement de fond. Par la suite, vous pouvez passer facilement à d'autres bases de données.

T. Effectuez-vous un contrôle des connaissances en fin de stage ?

G.L. Nous avons un premier problème : l'hétérogénéité des gens inscrits à un stage. Aussi, en accord avec nos clients, nous effectuons à l'inscription un petit contrôle des connaissances en informatique sous forme d'auto-test. Suivant les résultats, les gens choisissent tel ou tel type de stage dans un cursus qu'on leur a présenté. Ils peuvent se situer par rapport au niveau souhaité.

De plus en plus, nous sommes amenés à faire aussi un contrôle des connaissances en fin de stage pour s'assurer que les concepts de base ont été compris, par exemple sous la forme d'un petit exercice final. Cela n'a pas un caractère officiel. Mais cela permet d'améliorer le contenu de nos stages parce le recueil des opinions à la fin du stage mesure souvent plus l'atmosphère que les acquis de savoir-faire.

T. Avez-vous des expériences de reconversion de chômeurs ?

G.L. A la création de DEFI, nous avons réalisé pour la Direction du Travail une formation d'analystes-programmeurs Cobol en six mois. Il s'agissait de chômeuses ayant un niveau d'études entre la troisième et la terminale. Le stage a été sanctionné par un diplôme AFPA, dont l'examen a été réussi par les douze stagiaires. Toutes ont trouvé un emploi dans les deux ou trois mois après le stage. Dans ce type de formation, le niveau de départ des stagiaires s'apparente à celui des utilisateurs de micro dans les entreprises. Nous nous sommes tenus rigoureusement à notre pédagogie de résolution de problèmes et d'exigence de méthodologie.