



HAL
open science

Former un praticien réflexif de l'ingénierie du logiciel

Philippe Saliou, Vincent Ribaud

► **To cite this version:**

Philippe Saliou, Vincent Ribaud. Former un praticien réflexif de l'ingénierie du logiciel. 4ème Colloque Questions de Pédagogie dans l'enseignement supérieur, Jan 2007, Belgique. pp.535-545. hal-00504340

HAL Id: hal-00504340

<https://hal.univ-brest.fr/hal-00504340>

Submitted on 20 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FORMER UN PRATICIEN REFLEXIF DE L'INGENIERIE DU LOGICIEL

P. Saliou, V. Ribaud

EA3883, Département informatique, Université de Brest, France

E-mail : Philippe.Saliou@univ-brest.fr, Vincent.Ribaud@univ-brest.fr

Résumé

L'Université de Bretagne Occidentale développe depuis 5 ans un dispositif de formation innovant dont les principaux objectifs sont la maîtrise des activités d'ingénierie du logiciel, le travail coopératif en équipe et le développement des systèmes d'information. L'ambition majeure de ce dispositif est de former des praticiens qui sauront s'adapter aux changements et être force de proposition sur les modèles, méthodes et pratiques auxquels ils seront confrontés. Le champ se déplace alors de la méthode du problème de Dewey au praticien réflexif de Donald Schön.

Mots-clés :

Apprentissage par Immersion. Réflexivité. Ingénierie. Amélioration continue.

I INTRODUCTION

Cet article décrit, sous l'éclairage du rapport entre formation et expérience, la dernière année d'un Master en Ingénierie du Logiciel. La plupart des analyses de l'enseignement de l'ingénierie du logiciel mettent en avant l'apprentissage par problème et par projet, en équipe et à long terme (ACM-IEEE Joint Task Force, 2001), (Meyer, 2001). Cette pédagogie est mise en œuvre lors d'une année passée en immersion à l'université dans une entreprise virtuelle, sans cours traditionnels, lors de laquelle les étudiant(e)s travaillent en équipe pour concevoir et réaliser un système d'information, dans les règles strictes de l'ingénierie du logiciel et sous la conduite et le tutorat d'un professionnel expérimenté (Saliou & Ribaud, 2005).

Guy Le Boterf (2004) fait une distinction très éclairante entre "la" compétence du professionnel que l'on reconnaît à sa capacité à gérer un ensemble de situations professionnelles et "les" compétences (relationnelles, techniques, savoir-faire, ...) qu'il doit combiner et activer pour pouvoir gérer ces situations. La construction de compétences reste central dans un dispositif de formation, mais il faut aussi viser la formation d'expériences sans pour autant la déléguer entièrement au monde professionnel (stages, alternance, apprentissage). C'est cette somme d'expériences qui peut conduire à "la" compétence. On rejoint les préoccupations de John Dewey (1938) "I take it that the fundamental unity of the newer philosophy is found in the

idea that there is an intimate and necessary relation between the process of actual experience and education”.

Dans l’immersion, le fil conducteur de l’apprentissage est le projet. Le projet est un processus unique qui consiste en un ensemble d’activités coordonnées et maîtrisées, comportant des dates de début et de fin, entrepris dans le but d’atteindre un objectif conforme à des exigences spécifiques, incluant des contraintes de délais, de coûts et de ressources (Afnor X50-105, 1991).

A partir du projet, un processus guide l’apprentissage des activités ("des" compétences) d’ingénierie. Ce processus d’apprentissage, rythmé par le cycle de production du logiciel, se décompose en un enchaînement de séquences. Chaque séquence est organisée en fiches d’apprentissage et de production, individuelle ou collective, définissant précisément le travail à réaliser, les apprentissages visés et/ou "les" compétences à mobiliser, l’articulation avec les activités précédentes, les ressources pédagogiques, la forme et le contenu des produits attendus.

En réunifiant le lieu, le temps et les moyens, le système par immersion fournit de nombreuses situations d’apprentissage, génératrices d’expériences. Le dispositif d’apprentissage a la vocation d’être ce que Donald Schön appelle "a reflective practicum" qui se situe à l’opposé de ce qui est généralement admis comme un curriculum professionnel normatif tel que le définit Edgar Schein (1972).

Dans cet article, nous rappellerons les principes et les concepts de notre dispositif d’apprentissage par immersion. Ensuite, nous détaillerons les spécificités pédagogiques qui contribuent à préparer nos étudiant(e)s à devenir de futurs praticiens réflexifs.

II L’INGENIERIE DU LOGICIEL PAR IMMERSION

II.1 Description générale de la formation

Le dispositif pédagogique du Master Professionnel « Ingénierie du Logiciel par Immersion » reproduit autant que possible les phénomènes que les futurs ingénieurs seront amenés à rencontrer : un environnement de travail professionnel, la relation client-fournisseur, le respect d’un référentiel de développement, l’emploi de méthodes et des outils associés, la coopération entre membres d’une équipe, etc. L’année de formation est structurée en trois périodes appelées itérations : une période d’apprentissage tutoré de 4 mois, une période de mise en pratique accompagnée de 2 mois, une période de stage en entreprise de 4 à 6 mois.

Pendant les deux premières périodes (6 mois), les étudiant(e)s sont immergé(e)s dans un véritable projet qui se déroule exactement comme en entreprise, avec les spécificités suivantes :

- Les étudiant(e)s sont réparti(e)s par équipe de 6 appelée compagnie. Les étudiant(e)s doivent jouer plusieurs rôles : organisationnel, technique, soutien.
- Le projet se déroule au sein de l'université, dans une salle spécialement réaménagée, simulant l'environnement standard des entreprises de développement logiciel.
- L'enseignant qui pilote et tutore le projet, appelé tuteur de compagnie, est issu du monde professionnel. Il s'appuie sur un référentiel d'apprentissage spécifique.
- Une plate-forme technologique professionnelle est mise à disposition de chaque compagnie.
- Un processus d'apprentissage est intégré dans l'ensemble des processus de production sur lesquels s'appuie la fabrication d'un logiciel : Gestion de Projet, Ingénierie de Développement du Logiciel et Soutien au Développement.
- Chaque compagnie construit son système d'information à partir d'un cahier des charges différent, avec des processus similaires mais en utilisant chacune ses propres méthodes, technologies et outils.

II.2 Dispositif d'apprentissage

Un dispositif de formation axé sur la transférabilité des compétences impose une révision systématique des pratiques pédagogiques et évaluatives. Pour Jacques Tardif, l'organisation scolaire dresse des barrières difficilement franchissables, qu'elles soient curriculaires ou liées à l'organisation du travail. La division disciplinaire nuit considérablement à la transférabilité des connaissances et des compétences développées par les élèves en milieu scolaire et elle limite énormément les enseignants dans la création d'environnements axés sur le transfert des apprentissages. De plus, l'organisation du travail en milieu scolaire divise les tâches et les horaires des enseignants et des élèves comme s'il s'agissait de travailleurs spécialisés sur une chaîne de montage en fixant des lieux, des durées et actions ponctuelles pour chaque discipline (Tardif, 1999).

En réunifiant le lieu, le temps et les moyens (à l'image du projet du monde professionnel) et en soutenant l'acquisition des compétences par un processus d'apprentissage spécifique, notre dispositif de formation est une alternative à l'organisation habituelle des enseignements à l'université tout en satisfaisant à l'objectif de transférabilité des compétences en ingénierie du logiciel.

Le processus d'apprentissage est guidé par un cycle d'ingénierie du logiciel depuis sa commande par le maître d'ouvrage jusqu'à sa mise en exploitation.

- Il se matérialise sous la forme d'un enchaînement de séquences d'apprentissage avec des objectifs clairs et contrôlés à chaque séquence.
- A chaque séquence correspond un certain nombre d'activités décrites par des fiches d'apprentissage définissant précisément le travail à réaliser, les apprentissages visés et/ou les compétences à mobiliser.
- Chaque fiche d'apprentissage est affectée à un(e) ou plusieurs étudiant(e)s qui doivent alors exercer le métier (ou rôle) inhérent à l'activité concernée.
- Chaque séquence met à disposition un ensemble de fournitures pédagogiques que les étudiant(e)s doivent s'approprier.
- Pour chaque fiche d'apprentissage, les étudiant(e)s peuvent s'appuyer au quotidien sur l'assistance de tuteurs d'apprentissage.
- Chaque fiche d'apprentissage décrit la forme et le contenu des produits attendus. Elle constitue le support principal de l'évaluation.

La durée d'une séquence peut varier de 1 à 2 semaines en fonction des travaux mis en œuvre dans la séquence. Le parcours de formation comprend deux cycles (appelés itérations) du processus. Lors de la première itération, à chaque changement de séquence l'étudiant(e) se voit confier de nouveaux travaux qui correspondent à des changements sur plusieurs axes : domaine, activité, rôle, etc. Lors de la deuxième itération, une organisation fixe du travail est mise en place. Un(e) étudiant(e) assume la fonction de chef de projet, les autres assurant les autres métiers participant à la conception et à la fabrication d'un logiciel : analyste, concepteur, développeur, etc.

III UN PRACTICUM REFLEXIF

En accord avec Dewey et le modèle de l'Education nouvelle, l'apprentissage par immersion est un système éducatif qui procède du dedans et non une formation qui se fait du dehors. Les actions des étudiant(e)s sont à l'origine de tout apprentissage. Nous reprenons le modèle utilisé par Argyris et Schön (1974) pour expliquer leurs théories de l'action.

Les valeurs sous-jacentes (*governing variables*) sont les dimensions qu'une personne cherche à maintenir dans des limites acceptables. Les stratégies d'action (*action strategy*) sont les mouvements et les plans qu'une personne utilise pour conserver ses valeurs sous-jacentes dans l'intervalle acceptable. Les conséquences sont les résultats d'une action, intentionnelles ou non.

Pour Argyris et Schön (1978), apprendre nécessite la détection et la correction des erreurs. Quand quelque chose ne marche pas, la plupart des gens cherchent une autre stratégie qui fonctionnera en accord avec les valeurs sous-jacentes. C'est ce qu'ils appellent l'apprentissage à boucle unique. Une alternative est de questionner les

valeurs sous-jacentes elles-mêmes, de les soumettre à un examen minutieux et critique. Argyris et Schön parlent alors d'apprentissage à double boucle. Ceci peut conduire à une remise en question des valeurs et donc à un déplacement du cadre où opèrent les stratégies et surviennent les conséquences.

Ce questionnement permanent, qu'il soit avant, pendant ou après l'action, est central dans notre dispositif. Nous rejoignons les notions de réflexion en cours d'action et de réflexion sur l'action de Schön (1983), sans chercher cependant à établir une distinction entre les deux. Dans sa théorie, c'est surtout la notion de *practicum* réflexif (*reflective practicum*) qui nous paraît centrale.

[...] its main features are these. It's a situation in which people learn by doing, and I hope the ghost of John Dewey is circling just over my head. [...] Where they learn by doing in a *practicum* which is really a virtual world. A virtual world in the sense that it represents the world of practice, but is not the world of practice. A virtual world in the sense that, in that world, students can run experiments cheaply and without great danger. [...] And they learn by doing with others in the virtual world of the *practicum* in interaction with someone who is in the role of coach, more like a coach than like a teacher, because that coach is trying to help them do something (Schön, 1987b).

Nous considérons le dispositif d'apprentissage par immersion comme un *practicum* réflexif. La nature de l'expérience éducative de l'apprentissage par immersion est tout à fait différente d'autres systèmes éducatifs.

[...] the experience of the students in any *reflective practicum* is that they must plunge into the doing, and try to educate themselves before they know what it is they're trying to learn. The teachers cannot tell them (Schön, 1987b).

Cette section présente quelques points du système par immersion qui contribuent à l'établissement d'un *practicum* réflexif : la rétroaction continue du cycle auteur-tuteur, l'apprentissage coopératif par revue de pairs, un cycle d'ingénierie pour apprendre à faire suivi d'un autre cycle pour faire, l'ajustement d'activité, l'usage de la rétro-ingénierie.

III.1 Cycle Auteur-Tuteur

Le système d'apprentissage par immersion reposant sur une simulation de la vie réelle et sur la construction du savoir, l'évaluation prend place dans un contexte pédagogique qualifié d'authentique ou d'alternatif (Tardif, 1998).

Lors de la première période d'«Apprentissage tutoré», la fiche d'apprentissage est le support principal de la relation Tuteurs-Étudiants. Elle décrit le travail à réaliser, les différentes fournitures pédagogiques sur lesquelles doit s'appuyer l'apprentissage ainsi que les produits attendus. Le principe de l'évaluation formative est le suivant. Chaque fiche d'apprentissage donne lieu à un ou plusieurs produits de sortie. Le

tuteur évalue le(s) document(s) et produit(s) réalisé(s) : il annote, corrige, propose, réoriente. Il rédige une fiche d'évaluation qui comprend un bilan général de l'évaluation ainsi que la liste exhaustive des points à améliorer ou à refaire.

Lorsqu'ils prennent connaissance des fiches d'évaluation, les premières réactions des étudiant(e)s sont généralement défensives. Préoccupé(e)s de réussir et d'éviter l'embarras des sentiments négatifs (dû au feedback pris comme un jugement), leurs stratégies de réponse cherchent à se protéger et à contrôler unilatéralement le travail à refaire, en appliquant chaque remarque « à la lettre » et en sollicitant le tuteur seulement lorsque le feedback perçu comme une consigne n'est pas compris. De telles caractéristiques sont décrites dans le Modèle I d'Argyris et Schön (1974).

Lorsque l'étudiant(e) s'habitue au sentiment d'inconfort et de vulnérabilité lié au feedback et qu'il(elle) comprend que ses routines défensives limitent sa capacité à apprendre et à se développer, il devient possible d'approfondir, de discuter, de remettre en question collégialement, les remarques apportées par le tuteur. À l'issue de ce débriefing, les étudiant(e)s concerné(e)s doivent mettre à jour et même parfois refondre entièrement leur production. Il s'ensuivra une deuxième évaluation pour vérifier la prise en compte des remarques initiales. En général, deux évaluations sont suffisantes pour obtenir un résultat satisfaisant en termes d'apprentissage. On a alors fait un pas vers le Modèle II d'Argyris et Schön (1974), détaillé au III.2, qui est une théorie de contrôle et de recherche collaborative.

III.2 Revue de pairs

Dans l'industrie, le but des revues de pairs est de détecter et d'éliminer, précocement et efficacement, les défauts des produits au cours de leur développement. En effet, tout défaut non détecté au cours d'une phase induira sa propagation dans les autres produits du cycle de développement et un travail de reprise ultérieure.

Dans le cadre de notre formation, nous avons adapté et étendu la revue de pairs telle que pratiquée dans l'industrie pour nous en servir comme support aux transferts de compétences entre étudiant(e)s d'une même compagnie (Saliou & Ribaud, 2005). La revue de pairs devient désormais un vecteur de transferts de connaissances et de savoir-faire entre les différents co-apprenants.

Une telle revue de pairs privilégie davantage la collecte des données sur les interrogations, compléments d'information, propositions d'amélioration que la collecte des données sur les défauts potentiels même si celle-ci en fait implicitement partie. Ces données vont permettre de favoriser les échanges entre étudiant(e)s, de prendre en compte certaines propositions d'amélioration ainsi que de corriger les défauts majeurs dans les produits de sortie des apprentissages.

Si les premières valeurs sous-jacentes de cette coopération sont celles du Modèle II d'Argyris et Schön (1974) (l'information valide, le choix libre et informé, l'engagement interne), les premières stratégies d'action doivent combiner le

plaidoyer et l'investigation, rendre le raisonnement explicite et discutable et encourager les autres à faire de même. Il en résulte une capacité accrue à améliorer les stratégies définies pour atteindre les buts (apprentissage à boucle unique) mais aussi de choisir parmi les normes, les objectifs et les valeurs mis en concurrence (apprentissage à double boucle).

III.3 Processus d'Apprentissage/Production

Le parcours de formation comprend deux cycles (appelés itérations) du processus d'apprentissage. La première itération est fortement tutorée; on pourrait aussi la qualifier de « mentorée » ou « entraînée » (les termes anglais « mentoring » et « coaching » seraient plus significatifs). Elle est avant tout destinée à l'apprentissage et les étudiant(e)s la perçoivent très vite comme une occasion unique où il(elle)s ont droit à l'erreur (l'évaluation est avant tout formative).

Une organisation fixe du travail est mise en place pour la deuxième itération. L'objectif de la deuxième itération est la production et la mise en exploitation du système d'information complet tel que demandé dans le cahier des charges. Les étudiant(e)s vont s'appuyer sur les fiches d'apprentissage de la première itération pour mettre en œuvre (de manière autonome) un processus de développement du logiciel en phase avec les choix d'architecture technique, les outils de développements utilisés ainsi que les contraintes de délais imposés par le client.

Lorsque les étudiant(e)s s'interrogent sur la manière dont ils ont agi lors de la première itération tandis qu'ils sont en train de réaliser la seconde itération, on se situe dans la perspective de la réflexion sur l'action de Schön (1983). Les étudiant(e)s contribuent alors à la constitution d'un répertoire d'expériences (idées, exemples, actions, ...) sur lequel il(elle)s prendront appui pour traiter les situations nouvelles.

III.4 Ajustement d'activité

Dans le développement des systèmes informatiques, l'ajustement est une activité amont qui incombe au chef de projet et dont la finalité est d'adapter les processus standard d'une organisation pour les ajuster aux caractéristiques spécifiques de leur utilisation (SPICE, 1995). En d'autres termes, il s'agit d'ajuster le cadre référent des « bonnes pratiques » de sa société aux spécificités de son projet en termes de contraintes industrielles, d'exigences client, etc.

Dans notre processus d'apprentissage, nous transformons ce concept d'ajustement en un « ajustement d'activité » destiné à favoriser la réflexivité sur l'action. Le principe est que l'apprentissage de certaines activités d'ingénierie du logiciel relativement complexes s'appuie sur l'ajustement de l'activité correspondante.

L'ajustement d'une activité peut être assimilé à un travail préalable de réflexion et de proposition quant à la manière (pas forcément la meilleure) de procéder pour réaliser l'activité concernée. A titre d'exemple, l'utilisation d'une nouvelle méthode ou d'un nouvel outil débutera par son exploration afin d'en ajuster son usage aux caractéristiques spécifiques du projet. De même lorsqu'une tâche de production est difficile à appréhender comme la conception (qui nécessite une réelle expérience), il est préférable de réfléchir en amont en réalisant par exemple, la conception a posteriori d'un prototype de validation de l'architecture technique réalisée lors de la séquence d'apprentissage précédente.

Tout ajustement doit se solder par la production d'un guide d'usage ou de mise en œuvre de l'activité concernée. Cette notion d'ajustement d'activité favorise et encourage les initiatives et la créativité des étudiant(e)s sur les aspects techniques, méthodiques ou sur tout autre activité du processus de développement logiciel.

Lorsque le répertoire de l'étudiant(e) est vierge pour une activité donnée, toute nouvelle tâche semble impossible. L'ajustement d'activité permet d'initialiser le répertoire. Après l'ajustement, face à l'activité visée, même si elle est perçue comme nouvelle et unique, l'étudiant(e) voit qu'il y a quelque chose de présent et de familier dans son répertoire.

III.5 Rétro-Ingénierie

En informatique, la rétro-ingénierie d'un système d'information est le processus de reconstruction de la spécification fonctionnelle et technique d'un système opérationnel, principalement à partir du code des programmes. La reconstruction de ces spécifications a généralement pour objectif la documentation, la conversion, la restructuration, la maintenance ou l'extension du système d'information.

En pédagogie, la rétro-ingénierie est une démarche inductive. Il s'agit de reconstruire à rebours un processus en partant d'une activité.

[...] la démarche de la rétro-ingénierie, ou l'ingénierie intuitive [...] revient à analyser, décomposer pas à pas une activité afin de se demander comment elle est mise en œuvre dans une situation donnée. Cette démarche est pratiquée chaque fois qu'il est nécessaire de comprendre « comment ça marche » ou « comment c'est fait » (Dessus, 2000 citant Pinker).

Il s'agit de la même notion mais avec une différence fondamentale. Dans la rétro-ingénierie d'un système, le praticien en informatique est censé maîtriser les techniques (et donc les activités) qui ont concouru à sa réalisation.

Dans notre dispositif d'apprentissage, nous avons couplé ces deux points de vue pour en faire un support réflexif facilitant et accélérant l'acquisition en situation des connaissances et des savoir-faire de nos étudiant(e)s.

Le principe consiste à confronter les étudiant(e)s avec un système informatique existant dont ils ne connaissent ni les technologies ni les outils de développements

associés. L'objectif est d'acquérir un premier niveau de maturité qui leur permettra de maintenir et de faire évoluer ce système. La démarche s'appuie sur une trame standard :

1. Installer, faire fonctionner et étudier le système à l'aide des ressources pédagogiques mises à leur disposition.
2. Installer et configurer l'environnement de production qui leur permettra de modifier le système.
3. Réaliser une maquette répondant aux mêmes caractéristiques techniques avec une obligation permanente d'observer leurs pratiques de production.
4. Rédiger un document permettant aux autres membres de l'équipe de prendre en main et de poursuivre la maquette réalisée.

Pour prendre une analogie simple avec le Génie Mécanique, il est demandé aux étudiant(e)s d'analyser et de mettre en service une chaîne de montage (le système informatique) puis d'en construire un prototype. Ce prototype doit être réalisé de manière plus ou moins structurée à l'aide de leurs seules connaissances et des outils à leur disposition tout en s'interrogeant sur les méthodes et les outils qu'ils mettraient en œuvre pour réaliser une nouvelle chaîne de montage. Cette nouvelle chaîne devra répondre à de nouvelles exigences techniques et fonctionnelles.

La rétro-ingénierie d'une activité ajoute des expériences itératives et incrémentales au répertoire de l'étudiant(e). Le but est de faire prendre conscience qu'une situation perçue comme nouvelle ne l'est pas entièrement, qu'il(elle) peut établir des liens avec des situations familières, antérieures ou nouvelles, appartenant à son répertoire.

It is [...] to see the unfamiliar, unique situation as both similar to and different from the familiar one, without at first being able to say similar or different with respect to what. The familiar situation functions as a precedent, or a metaphor, or ... an exemplar for the unfamiliar one (Schön, 1983).

III.6 Quelle dimension pour la réflexivité ?

La particularité des supports « réflexifs » présentés dans cet article est qu'ils s'inscrivent systématiquement dans un contexte authentique d'apprentissage, sous la forme de pseudo-activités réflexives d'ingénierie.

Ces pseudo-activités réflexives d'ingénierie, même si elles sont intégrées dans le processus d'Apprentissage/Production ne sont pas des activités directes de production. Nous les considérons comme des activités d'apprentissage placées dans une dimension orthogonale qualifiée de « réflexivité continue » par analogie à « l'amélioration continue » prônée par l'ISO 9001:2000 pour améliorer les processus au sein de l'entreprise (Mitonneau, 2004).

Les étudiant(e)s ne perçoivent pas ces activités réflexives comme des activités d'ingénierie issues du monde professionnel, mais ils ne les perçoivent pas non plus comme un outil pédagogique contraignant dont la finalité serait d'être réflexif à tout

prix. Il(elle)s les ressentent comme des étapes rassurantes leur permettant de prendre du recul et d'asseoir leurs connaissances. Il(elle)s se les approprient rapidement car ils y voient un intérêt immédiat quant à la manière rationnelle d'aborder le problème auquel ils sont confrontés. Il(elle)s s'aperçoivent également que ces activités contribuent à améliorer leur propre processus d'apprentissage, indirectement leur processus de production et au final la qualité du système attendu.

Nous avons proposé que ce modèle d'action réflexive continue soit mis en pratique dans l'industrie sur des projets réels (Saliou & Ribaud, 2004). La « réflexivité continue » facilite l'adaptation des praticiens aux innovations et changements rapides des technologies de l'information.

V CONCLUSION

Pour surmonter les défis rencontrés dans leur pratique, les professionnels se fondent plus sur leur répertoire d'expériences et une certaine ingéniosité acquise au cours de leur pratique plutôt que sur des formules apprises au cours de leur formation fondamentale. D'où la nécessité d'initialiser et de développer un modèle de pensée et d'action réflexive dans leur formation initiale.

Le dispositif par immersion utilise un processus d'apprentissage et d'exploration qui est central dans un *practicum* réflexif : le savoir d'expérience s'y construit, entre autres, grâce à la pratique de l'exercice de l'analyse réflexive. Nous avons présenté dans cet article quelques points du dispositif qui favorisent cette pratique. L'objectif majeur est d'être capable, à terme, d'améliorer les modèles, méthodes et pratiques standards (normalisés ou non) en vigueur au sein de son entreprise, ce qui nécessite une remise en cause des valeurs sous-jacentes. Tous les étudiant(e)s n'atteindront pas ce niveau de maturité au cours de leur formation, le déclic pouvant se faire plus tard. Néanmoins, même si cet apprentissage à double boucle n'est pas acquis, le dispositif rend capable d'améliorer ses propres pratiques ainsi que celles de ses pairs (apprentissage à boucle unique).

Il reste autant de questions ouvertes qu'en pose le livre de Schön (1987a) comme :

- Comment les professionnels s'y prennent-ils pour résoudre les problèmes qui leur sont soumis ?
- Comment se forme la pensée du professionnel dans le feu de l'action ?
- Comment se développe cette virtuosité dont ils font preuve ?

RÉFÉRENCES

- ACM-IEEE Joint Task Force (2001). *Computer Curricula 2001*. USA, ACM-IEEE.
 Afnor X50-105 (1991). *Le management de projet*. Paris : Afnor.

- Argyris, C. & Schön, D. (1974). *Theory in practice: Increasing professional effectiveness*. San Francisco : Jossey-Bass.
- Argyris, C. & Schön, D. (1978). *Organizational learning : A theory of action perspective*. Reading, Mass. : Addison Wesley.
- Dewey, J. (1938). *Experience and Education*. New York : Simon & Schuster.
- Dessus, P. (2000). La planification des séquences d'apprentissage, objet de description ou prescription ? *Revue Française de Pédagogie*, 133, 101-116.
- Le Boterf, G. (2004). *Ingénierie et évaluation des compétences*. Paris : Les Editions d'Organisation.
- Meyer, B. (2001). Software Engineering in the Academy. *IEEE Computer*, 34, 28-35.
- Mitonneau, H. (2004). *ISO 9000 Version 2000*. Paris, Dunod.
- Saliou, P. & Ribaud, V. (2005). Ingénierie du logiciel par immersion. *Revue Génie Logiciel*, 74, 28-36.
- Saliou, P. & Ribaud, V. (2004). *Continuous tailoring activities in software engineering*. European Software Process Improvement Conference, Nov. 2004.
- Schein, E. (1972). *Professional Education: Some New Directions*. New York : McGraw-Hill.
- Schön, D. (1987a). *Educating the Reflective Practitioner: Toward a New Design for Teaching and Learning In the Professions*. San Francisco : Jossey-Bass.
- Schön, D. (1987b). *Educating the Reflective Practitioner*. Meeting of the American Educational Research Association, 1987.
- Schön, D. (1983). *The reflective practitioner*. New York : Basic Books.
- Software Process Improvement and Capability determination (SPICE) (1995). *Software Process Assessment - Version 1.00*. London : ISO/IEC.
- Tardif, J. (1998). *Intégrer les nouvelles technologies de l'information – Quel cadre pédagogique ?*. Paris : ESF.
- Tardif, J. (1999). *Le transfert des apprentissages*. Montréal : Les Editions LOGIQUES.