



Ingénierie du logiciel par immersion: allers-retours entre apprendre et faire

Philippe Saliou, Vincent Ribaud

► To cite this version:

Philippe Saliou, Vincent Ribaud. Ingénierie du logiciel par immersion: allers-retours entre apprendre et faire. 3e colloque SRED 2007, Sep 2007, France. pp.x-y, 2007. <hal-00504344>

HAL Id: hal-00504344

<http://hal.univ-brest.fr/hal-00504344>

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.

Ingénierie du logiciel par immersion: allers-retours entre apprendre et faire

Philippe Saliou, Vincent Ribaud

EA3883, LISyC, Université de Bretagne Occidentale, C.S. 93837 29238 Brest Cedex 3, France

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

Résumé

1 Introduction

Maîtriser une discipline est une tâche difficile. La plupart des professionnels diront que l'apprentissage et la maîtrise se sont développés « en faisant ». Ainsi, on constate l'emploi de plus en plus fréquent d'une « pédagogie par projet » destinée à pallier à certaines faiblesses de l'enseignement « classique » ou tout au moins la délégation d'une partie des études sous la forme de projets réalisés par les élèves et en dehors du temps des enseignements.

Quel est alors le rôle des enseignants dans cette pédagogie de projets ? Pour la plupart, leur activité se restreint à l'évaluation de ce qui se passe dans le projet, mais avec quels critères ? L'évaluation est le plus souvent celle du produit réalisé donc on peut supposer qu'elle obéit à des critères de production comme la conformité aux exigences du cahier des charges, la maîtrise des coûts et des délais.

Pourtant dans le monde de l'éducation, on devrait autant évaluer les progrès de l'apprentissage que la qualité des résultats. Ceci nécessite de laisser aux apprentissages le temps nécessaire, c'est-à-dire de libérer suffisamment de temps pour pouvoir accumuler de l'expérience, qui provient autant des succès que des erreurs. L'erreur (et le temps d'en faire) n'est pas préjudiciable à l'apprentissage alors qu'elle l'est à la production. Dans l'apprentissage, la capacité à réaliser un produit conforme à un cahier des charges n'indique que le moment où on est passé dans le monde de la production.

Cela semble indiquer qu'il existe des temps différents dans les deux mondes : apprendre et faire, avec un impact de ces temps sur les performances attendues. Apprendre et faire, apprendre en faisant, faire en apprenant, qu'importe la combinaison pourvu qu'elle existe. Cependant, il ne faut pas évaluer l'un avec les critères de l'autre, ni confondre le temps du faire et de l'apprendre.

Depuis Septembre 2002, l'U.B.O. offre un Master Pro 2ème année « Ingénierie du Logiciel par Immersion », où les étudiant-es passent les deux tiers de leur dernière année d'étude en entreprise virtuelle. L'idée est de leur faire vivre le déroulement d'un projet de A à Z, sur fond d'un référentiel d'assurance qualité ISO9001, des méthodes et outils associés aux architectures Web actuelles, mais dans des conditions d'apprentissage. En dehors des cours de langues et de communication, aucun cours traditionnel n'est dispensé. Les étudiant(e)s, par équipe de 6, sont immergé(e)s dans un vraisemblable (plutôt que véritable) projet. Chaque activité d'ingénierie se décompose en tâches. Chacune des tâches, réorientée apprentissage, devient alors, avec l'aide d'enseignants-tuteurs professionnels, l'occasion d'acquérir en situation, les connaissances et compétences en ingénierie des Systèmes d'Information. Le processus d'apprentissage s'appuie sur les processus de production de la société de services informatiques Thales Information Systems.

L'année de formation est structurée en trois périodes : 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. Lors de chaque période, le cycle de production, découpé en étapes puis en tâches, rythme les activités.

La première période permet l'apprentissage « en faisant » des connaissances nécessaires à chaque étape du cycle d'ingénierie grâce au tutorat quotidien, la deuxième période a pour but la production d'un système d'information qualifié et approuvé par le client. La troisième période, opérationnelle en entreprise, peut être l'occasion de prendre conscience des allers-retours entre l'apprentissage et la production.

La première période a un planning prévisionnel de 18 (16 ± 2) semaines, décomposé en 8 séquences d'apprentissage d'activités (d'1 à 3 semaines). Grâce à un volant de 2 semaines supplémentaires, le temps de chaque séquence est ajusté au rythme des apprentissages. Chaque tâche prend place dans un des processus de production de l'entreprise simulée. Les élèves perçoivent très clairement ce temps d'apprentissage comme un temps précieux « où on a le droit à l'erreur », mais il(elle)s ne le ressentent pas uniquement comme un temps d'apprentissage puisqu'ils apprennent également à produire. La plupart des tâches réalisées sont donc du ressort de l'apprentissage de la production. Les autres tâches appartiennent à une autre dimension qualifiée d'ajustement continu, très proche de la notion « d'amélioration continue » prônée par l'ISO 9000:2000 pour améliorer les processus au sein de l'entreprise à la différence essentielle qu'elle fait partie intégrante des processus de production.

Lors de la deuxième période, les élèves s'appuient sur le processus d'apprentissage pour mettre en œuvre (de manière autonome) un processus de développement logiciel permettant la production du système. Lors de cette itération, l'équipe reste toujours accompagnée et encadrée par le tuteur qui a alors plutôt une fonction de surveillance et de sauvetage.

La deuxième période a un planning de 8 semaines, non ajustable. Les élèves passent alors spontanément dans un rythme de production et mettent en œuvre les méthodes et outils de développements appris tout en respectant les contraintes de délais imposés par le client.

La question centrale posée par cet article est la prise de conscience et la pratique de fréquents allers-retours entre apprentissage et production, dans le système par immersion comme tout au long de la vie professionnelle. La section 2 présente un même processus pour l'apprentissage et la production. La section 3 compare l'évaluation dans les deux mondes. La section 4 s'interroge sur les pratiques qui favorisent les allers-retours entre apprendre et faire.

2 Un même processus pour l'apprentissage et la production

2.1 Activités d'un ingénieur du logiciel

2.1.1 *Modèle des activités d'une société de services en informatique*

Les *activités* d'ingénierie du logiciel généralement admises dans une société de services sont les suivantes :

- La *capture des exigences* a pour but de recueillir, d'établir et de formaliser les exigences du client sur le Système d'Informations à concevoir.
- L'*analyse* consiste à transformer les exigences fonctionnelles en les affinant et en les structurant, de manière à obtenir une idée de ce que va réaliser le système en terme de métier.
- L'*architecture technique* a pour finalité la modélisation et la description de l'organisation et de l'inter-fonctionnement des composants majeurs du système et l'élaboration de solutions techniques génériques satisfaisant aux exigences du client et garantissant l'évolution du système dans le temps.
- La *conception* vise à établir les capacités du produit et son architecture, ce qui comprend la décomposition du produit, l'identification des composants, les états et les modes du système, les interfaces majeures entre composants et les interfaces externes du produit.
- Le *codage* et les *tests unitaires* ont pour but de produire un ensemble d'unités testées indépendamment et destinées à être assemblées. Cet ensemble d'unités implémente la décomposition en composants issue de la conception.
- L'*intégration* a pour objet d'assembler les unités de réalisation du logiciel afin de produire le système qui satisfait aux exigences.
- Les activités de *tests* permettent de vérifier la conformité d'un logiciel par rapport à son référentiel et particulièrement garantir qu'il vérifie l'intégralité des exigences spécifiées.
- La *documentation utilisateur* a pour but d'écrire et de maintenir les documents nécessaires aux utilisateurs et aux administrateurs du système.
- L'*installation* et le *déploiement* ont pour but de préparer la livraison du système, de le livrer et de l'installer sur le site du client de manière à assurer son fonctionnement opérationnel.
- Lal consiste en un ensemble de processus qui établit le projet, coordonne et gère les activités et les ressources afin de produire le logiciel satisfaisant aux exigences du client.
- L'*assurance qualité* a pour objectif de s'assurer que les produits et les activités respectent les standards, les procédures et les exigences définies dans le plan de développement du logiciel.
- La *gestion de configuration* est une discipline de management de projet qui permet de définir, d'identifier, de gérer et de contrôler les articles de configuration tout au long du cycle de développement.
- Le *support technique* assure l'installation, l'administration, le suivi de l'exploitation et l'optimisation de l'ensemble des moyens et des ressources techniques (matériel, logiciel et réseau) nécessaires à l'équipe projet pour réaliser le système.

2.1.2 *Structure de la formation*

Dans le monde du travail, ces activités sont d'abord mises au service de la réalisation de l'affaire¹. Une affaire est décomposée en lots de travaux tenant compte des produits à livrer, des activités à réaliser et de l'organisation à mettre en place. Une activité se décompose en *tâches* et la capacité à effectuer une tâche peut s'appeler une *compétence*.

Dans un organisme, les activités sont regroupées en *familles*, qui forment souvent la structure de l'organisation : management, études, soutien, exploitation, ...

Notre structure d'activités est donc organisée comme suit :

- les activités se répartissent en 3 domaines principaux (les familles) qui peuvent être assimilés à des sous-processus de fabrication² et donc à des spécialisations du métier d'ingénieur logiciel (cf tableau 1) ;
- les activités et leurs tâches (donc les compétences mobilisées) sont situées dans un projet réaliste, fourni en deux lots et réalisé lors de deux périodes (appelées *itérations*).

L'organisation de la première période (16 ± 2 semaines) reproduit la structure des activités et cette décomposition se retrouve dans le curriculum informatique de la première période : les UE - Unités d'enseignements - (modules) correspondent aux domaines et les EC - Eléments Constitutifs - (sous-modules) aux activités.

L'organisation des activités du deuxième lot est déléguée aux élèves : à elles et à eux d'adapter la structure du 1^{er} lot (établi lors de la 1^{ère} itération) aux contraintes de production du 2^{ème} lot, donc de mobiliser leurs compétences à bon escient et au bon moment.

2.1.3 *Cadre de référence*

Le découpage des apprentissages en domaines et activités d'ingénierie forme le cadre de référence satisfaisant à la fois aux exigences de la branche professionnelle (comme décomposition du métier), aux règles de l'université en tant qu'institution (comme décomposition du curriculum en unités de cours), aux objectifs de l'équipe pédagogique ainsi qu'aux attentes des étudiants (comme nomenclature de référence de tous les apprentissages).

Le tableau ci-dessous donne la décomposition des EC (activités) dans les UE (domaines).

¹ Une affaire est l'ensemble des fournitures et des prestations (travaux ou services) définies par un contrat ou une commande.

² Il existe un quatrième domaine ou sous-processus qui est diffus dans toutes les activités : la relation client-fournisseur.

UE / Domaines	EC / Activités	ECTS
Langue et Techniques d'Expression (6 ECTS)	Anglais	4
	Techniques d'Expression	2
Gestion de Projet Logiciel (5 ECTS)	Conduite De Projet Logiciel	2
	Assurance Qualité Logiciel	1
	Gestion de Configuration du Logiciel	2
Ingénierie de Développement Logiciel (13 ECTS)	Capture Des Besoins	2
	Architecture Technique	3
	Analyse	2
	Conception du Logiciel	2
	Codage - Tests Unitaires	2
	Intégration – Qualification	2
Soutien au Développement Logiciel (6 ECTS)	Support Technique	1
	Support Méthodes et Outils	2
	Documentation Utilisateur	2
	Installation - Déploiement	1

Tableau 1 : Décomposition de référence

2.2 Un référentiel d'apprentissage mais aussi de production

2.2.1 Structure d'un référentiel ISO 9001 d'ingénierie du logiciel

Dans l'industrie, en simplifiant à l'extrême, un système de management de la qualité de type ISO 9001:2000 pourrait se définir comme un ensemble de processus, un processus étant défini comme un ensemble d'activités corrélées ou interactives qui transforme des éléments d'entrée en éléments de sortie. Pour une société de services et d'ingénierie en informatique (qui est le type d'employeur de nos diplômés), ses métiers sont de concevoir et de vendre ; de développer ; de maintenir des logiciels. On trouvera donc dans le référentiel d'ingénierie d'une SSII la cartographie et la définition des processus de vente, conception, fabrication, maintenance ; la description des activités et des interactions entre processus.

« TEMPO est le référentiel de la société Thales Information Systems, base d'une culture partagée et moyen de capitalisation des bonnes pratiques. C'est un ensemble de procédures, guides et notes définissant l'organisation et les activités régissant le fonctionnement de la société, la conduite d'affaire, le développement logiciel et l'intégration de système. » (Thales Information System, 2002).

Les documents du référentiel TEMPO sont organisés en niveaux. Au sommet se trouve le Manuel Assurance Qualité (MAQ), puis le niveau procédures, et enfin le niveau instructions, documents techniques détaillant la mise en œuvre des procédures.

Les procédures sont la base de la description d'un processus et des activités associées. Afin d'avoir la plus grande stabilité possible, elles abordent le QUOI, fait QUAND et PAR QUI. La procédure fixe les exigences sur le processus. Les procédures sont applicables, mais afin de répondre aux spécificités des affaires, chaque procédure donne des éléments d'ajustement du processus. Les Instructions, le plus souvent appelées " Guide ", sont des recommandations de mise en œuvre qui traitent du COMMENT et du AVEC QUOI. En règle générale, chaque instruction est rattachée à au moins une procédure. Certaines instructions sont applicables, certaines ne sont que d'utilisation conseillée.

2.2.2 Le référentiel d'apprentissage d'ingénierie du logiciel par immersion

Notre objectif a été de définir un référentiel devant répondre aux attentes de tous les « co-apprenants » que sont les élèves et les enseignants mais aussi de tous les autres intervenants impliqués dans la formation.

Les deux premières années, nous avons élaboré une première version d'un référentiel d'apprentissage du métier d'ingénieur logiciel en nous appuyant sur 3 choix pédagogiques essentiels :

- L'emploi de procédés recouvrant tous les aspects de production d'un logiciel (Jacobson & Booch & Rumbaugh, 1999).
- L'utilisation du référentiel³ ISO 9001 de la filiale informatique du groupe Thales (Thales Information System, 2002).
- Une évaluation enchâssée dans l'apprentissage (Tardif, 1993).

A l'issue de ces cinq années d'expérience, nous disposons d'un référentiel qualifié d'apprentissage- production. Ce référentiel soutient le processus d'apprentissage et de production mis en œuvre lors des deux premières itérations.

Le référentiel comprend à ce jour :

- Un référentiel métier TEMPO-ILI comprenant :
 - les éléments issus de l'ajustement du référentiel Thales-IS,
 - des références à d'autres référentiels de type ISO 9001,

³ Grâce à un accord passé avec le groupe Thales, nous disposons à des fins pédagogiques d'une partie du référentiel TEMPO et notamment des procédures et des guides relatifs à l'ingénierie du logiciel.

- des exemples de documents produits sur des projets réels.
- Des supports d'utilisation des plates-formes techniques :
 - guides d'installation et d'utilisation,
 - white papers et red books.
- Des ouvrages ainsi que des supports d'auto-formation.
- La description du dispositif d'apprentissage constitué principalement :
 - des fiches d'apprentissage définissant précisément le travail à réaliser.
 - des fournitures pédagogiques mises à disposition au fur et à mesure du projet.
 - des procédures d'évaluation des apprentissages.

Chaque activité d'ingénierie se décompose en tâches. L'apprentissage d'une tâche est décrit dans une fiche d'apprentissage. Elle décrit le travail à réaliser, les différentes fournitures pédagogiques sur lesquelles doit s'appuyer l'apprentissage ainsi que les produits attendus.

L'élève perçoit la fiche d'apprentissage comme un élément essentiel d'organisation de ses compétences. Par exemple, au sein du domaine d'ingénierie du développement, l'activité « capture des besoins » se décompose en capture des besoins fonctionnels, capture des besoins techniques et consolidation des exigences. Pour chaque tâche, l'élève peut associer les produits amonts issus d'activités précédentes, les guides de bonnes pratiques ayant soutenu la tâche, sa place dans le processus de production, les produits de sortie de la tâche, etc.

Lors de la deuxième itération, les étudiant(e)s vont s'appuyer sur le référentiel d'apprentissage de la première itération. Ce référentiel d'apprentissage va alors être utilisé comme un référentiel de production du logiciel.

2.3 Une unité de lieu, de temps et de moyens

2.3.1 Processus d'apprentissage et de production

Le fil conducteur de l'organisation est le projet⁴. Définition de la norme ISO 10006 (2003) reprise par l'AFNOR X50-105 : « 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 [X50-105]. »

En réunifiant le lieu, le temps et les moyens (à l'image du projet/affaire du monde professionnel), le système par immersion fournit un processus d'apprentissage. Ce processus est guidé par le cycle de développement logiciel retenu et possède les caractéristiques suivantes :

- 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 de tâches décrites par des fiches d'apprentissage établissant les apprentissages visés et/ou les compétences à mobiliser.
- Chaque fiche d'apprentissage est affectée à un ou plusieurs étudiant(e)s qui doivent alors assumer le métier (ou rôle) inhérent à l'activité concernée.
- Chaque séquence met à disposition des 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 (qui est, la plupart du temps, le tuteur de compagnie).
- Chaque fiche d'apprentissage décrit la forme et le contenu des produits attendus. Elle constitue le support principal de l'évaluation.

2.3.2 De l'apprentissage à la production

Le parcours de formation comprend deux cycles (appelés itérations) du processus.

- 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. Chaque tâche donne lieu à une ou plusieurs fournitures (« deliverable »). Chaque fourniture est soigneusement examinée et annotée par le(s) tuteur(s), puis un « feedback » est restitué aux auteurs de la fourniture assorti des améliorations à apporter. Ce processus est réitéré (au moins 2 fois) jusqu'à ce que la fourniture soit jugée suffisamment satisfaisante pour pouvoir être exploitée par la suite (il peut y avoir parfois un problème lorsque la fourniture finale n'est pas assez satisfaisante). Les élèves perçoivent très vite cette itération 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 élèves vont s'appuyer sur les fiches d'apprentissage de la première itération pour mettre en œuvre (de manière autonome) un processus de développement logiciel en phase avec les choix d'architecture technique, les outils de

⁴ L'affaire est le terme employé dans le paragraphe précédent, mais il signifie un projet défini dans le cadre d'un contrat.

développements utilisés ainsi que les contraintes de délais imposés par le client. La deuxième période a un planning de 8 semaines, non ajustable. Les élèves passent à mettre en œuvre les méthodes et outils de développements appris tout en respectant les contraintes de délais imposés par le client. L'évaluation porte sur les produits réalisés selon des critères de production : conformité et délais.

- Lors de la deuxième période, les élèves s'appuient sur le processus d'apprentissage/production de la première période pour mettre en œuvre (de manière autonome) un processus de production/apprentissage du système. L'équipe reste toujours accompagnée par le tuteur qui a alors plutôt une fonction de surveillance et de sauvetage. Les élèves passent alors spontanément dans un rythme de production. Cependant, il(elle)s emploient naturellement des activités d'ajustement ou d'exploration telles que pratiquées dans la première période. Bien qu'étant sous la pression du temps, les élèves ont intégré cette dimension d'ajustement continu comme vitale et ne la sacrifient pas à la productivité.

3 Evaluation

3.1 Evaluation des processus de production dans l'industrie

Le cycle d'ingénierie du logiciel part de la commande du client jusqu'à la mise en service par le fournisseur. Le cahier des charges (ou la première phase du projet) doit établir un ensemble d'exigences fonctionnelles (qui décrivent des services du produit) et exigences techniques (qui énoncent des contraintes). Les exigences établissent un contrat entre le client et le fournisseur.

L'approche processus permet de décrire l'organisation d'une entreprise dans son aptitude à produire des résultats définis. Les processus les plus évidents sont ceux mis en œuvre pour réaliser les produits et services fournis par l'organisme, qu'on appelle souvent les processus opérationnels. Dans le domaine du logiciel, la chaîne des activités de production fournit à chaque étape intermédiaire des produits de sortie qui sont le plus souvent des documents. La description des activités comme la rédaction générique des résultats documentaires issus des activités forme une partie du référentiel du système de management de la qualité.

L'application du système de management de la qualité doit :

- garantir la conformité au référentiel d'action (approche centrée sur les activités et les procédures),
- mettre l'accent sur les résultats à atteindre et la satisfaction du client (approche centrée sur le produit).

Ces deux approches ne sont pas exclusives car elles concourent au même but. Elles sont toutes deux soutenues par deux catégories d'évaluations : celles inspectant les produits en cours d'élaboration et celles vérifiant et validant les exigences.

3.1.1 Activités d'inspection

On peut trouver plusieurs types d'activités d'inspection des produits en cours d'élaboration : lecture croisée, revue de pair, contrôle de processus. Dans tous les cas, ce qui est visé est la détection et la correction des défauts des produits en cours d'élaboration.

3.1.2 Activités de vérification et validation

Lorsqu'on évalue le respect des exigences, on peut distinguer vérification et validation. La vérification permet de déterminer si les produits de sortie d'une phase satisfont aux exigences établies pendant les phases précédentes. La validation évalue le produit lorsqu'il est fini pour s'assurer de sa conformité avec les exigences.

3.2 Caractéristiques de l'évaluation des apprentissages

3.2.1 Cadre théorique

Jacques Tardif écrit que les retombées du paradigme constructiviste sur les pratiques évaluatives sont nombreuses et diversifiées. [...] En conformité avec les pratiques d'enseignement, les évaluations reposent nécessairement sur des tâches complètes, complexes et signifiantes. [...] Les stratégies cognitives et méta-cognitives de l'élève doivent être prises en compte directement et explicitement. [...] Le contenu retenu lors des démarches d'évaluation doit être assez familier pour l'élève, de sorte que ce dernier puisse établir des liens entre ce contenu et ses connaissances antérieures. [...] Les évaluations doivent systématiquement rendre compte de l'état de l'organisation et de la hiérarchisation des connaissances par l'élève [Tar93].

3.2.2 L'enchâssement de l'évaluation dans l'apprentissage

Dans le cadre du paradigme constructiviste, l'évaluation est enchâssée dans les situations d'apprentissage et fait partie de l'environnement pédagogique par immersion. Il n'y a pas de séparation entre l'apprentissage et l'évaluation. Les étudiant(e)s sont constamment en train de construire des connaissances et de développer des compétences. Le déroulement de ces constructions et de ces développements est observé très régulièrement par les tuteurs pour établir la base des rétroactions et des évaluations.

L'évaluation est intégrée à la démarche de soutien des apprentissages et toutes les informations résultant de l'évaluation renseignent les étudiant(e)s sur l'évolution de leurs apprentissages et l'état de leurs connaissances. La distinction entre évaluation formative et évaluation sommative ne s'applique pas mais on établit deux types d'évaluation. Le premier type soutient directement la construction de savoir car il fournit une rétroaction continue aux apprenants. Il est appelé évaluation de régulation en référence à De Ketele : « processus ouvert dont la fonction prioritaire est d'améliorer le fonctionnement [...] d'une partie ou de l'ensemble du système. » [Ket93]. Le deuxième type est composé d'activités de vérification et de validation réalisées lors de jalons programmés.

3.2.3 *Evaluations de régulation*

Le dispositif par immersion cherchant à reproduire les processus de l'industrie, des éléments de forme de ces évaluations sont inspirés des activités d'inspection du §3.1.1. L'objectif étant l'apprentissage de ces processus, l'esprit est totalement différent : ces évaluations de régulation sont destinées à soutenir les apprentissages et principalement axées sur l'étayage et le désétayage (Tardif, 1998). En situation d'étayage, les rétroactions du tuteur visent à faire émerger les questions et/ou à fournir des indications et des suggestions permettant de construire le savoir manquant. En cas de difficulté, le tuteur consolide : il ré-explique, suggère une entraide, modifie la planification. Lorsque les élèves sont en mesure de réutiliser des connaissances et des compétences, le tuteur réduit le soutien et ses rétroactions visent essentiellement à l'amélioration des résultats ; à l'occasion il devient un apprenant (néanmoins en position de constructeur de savoir).

3.2.4 *Evaluations de vérification et de validation*

Ces évaluations s'inspirent des activités de vérification et de validation du §3.1. Elles contribuent au jeu de rôles qui est une des composantes de l'immersion : la simulation des jalons industriels de contrôle, tout en gardant l'aspect formatif de l'expérience car le tuteur donne des rétroactions pendant et après l'action.

3.2.5 *Des apprentissages aux produits*

Tout au long de la 1^{ère} période, les évaluations sont du 1^{er} type. Afin de respecter les caractéristiques énoncées dans le paragraphe précédent, l'évaluation ne peut porter que sur la mise en œuvre des tâches complexes d'ingénierie du logiciel. Une hypothèse importante est que l'évaluation de ces tâches complexes est en partie remplacée par une évaluation continue des éléments de sortie attendus. Toutes les évaluations contribuent à faire émerger les problèmes et à aider à construire les savoir-faire et les connaissances manquants. La situation d'apprentissage, complètement délimitée par le cadre de la fiche d'apprentissage et placée dans un modèle tâche-activité-domaine, aide l'élève et le tuteur à déterminer sur quelles capacités portent les apprentissages. A la fin de la 1^{ère} période, le lot (le produit) est qualifié par une évaluation du 2^{ème} type.

Lors de la 2^{ème} itération, les évaluations sont toutes du 2^{ème} type.

4 Allers-retours entre apprendre et faire

4.1 Favoriser la réflexivité

Le système d'apprentissage par immersion met en œuvre plusieurs dispositifs qui contribuent au développement d'un praticien réflexif au sens de Donald Schön: 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 [SR07].

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. 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.

4.2 L'ajustement continu d'activités

La plupart des tâches réalisées sont donc du ressort de l'apprentissage de la production. Les autres tâches appartiennent à une autre dimension qualifiée d'ajustement continu. Il existe une certaine catégorie d'apprentissages qui nécessitent d'être adaptés à la difficulté, aux connaissances antérieures, au processus de construction de connaissances de l'élève, ...

Ceci introduit une dimension supplémentaire dans les interactions entre processus que nous avons qualifié d'ajustement d'activité. L'ajustement d'activité est un travail de réflexion et de proposition sur la manière de réaliser une activité. Tout ajustement se solde par la production d'un guide d'usage ou de mise en œuvre de l'activité concernée. Le guide (qui correspond le plus souvent à une production méta-cognitive) est évalué. L'activité complexe qui en résulte peut alors avoir lieu et être évaluée sur ses productions de sortie.

Ce type de tâche est souvent planifiée en amont d'une tâche de production difficile à réaliser, comme la conception (qui nécessite une réelle expérience de fabrication) ou la mise en œuvre d'une technologie nouvelle ou complexe (qui nécessite un apprentissage) ; ces tâches portent généralement le nom de « ajustement de » (la conception, ...) ou « exploration » (de l'outil X, de la méthode Y). Cette notion d'ajustement continu est très proche de la notion « d'amélioration continue » prônée par l'ISO 9000:2000 pour améliorer les processus au sein de l'entreprise à la différence essentielle qu'elle fait partie intégrante des processus de production.

4.3 Suivi personnel de compétences

4.3.1 Définition des compétences de la formation

L'écriture du supplément au diplôme a été l'occasion de définir les compétences, savoir-faire et connaissances visés pour le Master d'ingénierie du logiciel par immersion. Le cadre de référence de la formation repose sur une décomposition du métier visé (ingénieur du logiciel) en domaines d'activités (comme la gestion de projet) et en activités à l'intérieur d'un domaine

(comme l'assurance qualité ou la conduite de projet). Pour chaque activité, nous avons défini l'ensemble des connaissances nécessaires ; une à une, les capacités ou savoir-faire liés à l'activité ; dans certains cas, les compétences-clés de l'activité.

A l'issue d'une analyse détaillée sur le métier (3 domaines, 14 activités, 48 capacités) et sur les compétences transversales (11), une synthèse (4 pages) a été produite et diffusée.

4.3.2 *Modèle de compétences*

Un modèle de compétences (ou ontologie de compétences) définit et contient la décomposition des informations. Un modèle de compétences est utilisé pour définir les éléments d'un curriculum (ou d'un référentiel) et leur relations. Pendant la période de formation, l'ensemble des élèves utilise le même modèle mais peut changer individuellement ensuite.

Un modèle est décomposé en domaines de compétences ; chaque domaine correspondant approximativement à une des grandes divisions de l'emploi, du métier ou de la fonction visé. Par exemple, un modèle de compétences pour un institut d'enseignement à distance pourrait contenir les domaines suivant : concepteur de cours, médiatiseur, instructeur et administratif.

Chaque domaine organise les compétences en familles de compétences. Une famille correspond à peu près à une activité principale du domaine. Ainsi, le rôle d'instructeur requiert des compétences qui pourraient être organisées dans les familles : adaptation au public, exécution du cours, évaluation, feedback.

A chaque famille est associée un ensemble de compétences, capacités ou savoir-faire, connaissances ; chaque entité est représentée par un libellé et une description.

4.3.3 *Autoévaluation et positionnement de niveaux de maturité*

Décidés à utiliser les résultats de l'analyse détaillée, nous avons élaboré le *Recueil Individuel de Compétences (RIC)* de la formation en « Ingénierie du Logiciel par Immersion », structuré selon le modèle présenté dans le § précédent et mis en œuvre dans l'outil de gestion des compétences eComp@s.

Les élèves établissent un Recueil Individuel de Compétences (RIC) en début d'année et après chaque itération. Pour chacune des capacités ou compétences transversales, chaque élève se positionne par rapport à un niveau de maturité. Les compétences transversales et les capacités sont évaluées sur une échelle de 1 à 5. L'échelle de positionnement des capacités est la suivante : 1 - Brouillard : Vague idée (voire pas d'idée du tout) ; 2 - Notion : A des notions, une culture générale mais insuffisantes pour sa mise en œuvre opérationnelle ; 3 - Utilisateur : Est capable de mettre en œuvre la capacité avec l'appui d'un collaborateur expérimenté et a une première expérience de sa mise en œuvre ; 4 - Autonome : Est capable de travailler de manière autonome ; 5 - Expert : Est capable d'agir en expert pour modifier, enrichir et développer la capacité ou le savoir-faire.

Au moment de l'établissement des quatre RIC, il-elle peut prendre alors conscience des objectifs de compétences affichés et positionner son niveau de maturité pour chaque capacité. A l'intérieur d'une itération, chaque fin de séquence est l'occasion pour l'élève de faire un point intermédiaire (et partiel) sur les acquisitions de la séquence et les compétences développées. Un journal permet de recueillir au fil de l'eau des commentaires sur chacune des capacités ou compétences transversales développées.

4.4 **Perspectives sur la gestion des compétences**

La majorité des systèmes de gestion de compétences comportent des outils d'évaluation utilisés pour identifier et estimer les compétences. Dans la perspective d'une autoévaluation, il est nécessaire de compléter l'autoévaluation d'un carnet de bord ou journal. Appliqué au développement des compétences, les buts possibles pour un portfolio incluent l'évaluation et la démonstration de l'atteinte des objectifs de compétences, l'illustration du processus d'apprentissage et de maîtrise de cet objectif... La possibilité d'écrire une entrée dans le journal pour chaque autoévaluation et d'y associer les productions essentielles de l'apprenant est un premier pas permettant de doter le système de suivi de compétences de quelques fonctions d'un portfolio. On peut penser qu'à terme une intégration plus poussée serait nécessaire mais le suivi personnel de compétences est plus facile à maintenir tout au long de la vie qu'un portfolio de compétences.

Utiliser le suivi de compétences et les portfolios pour l'évaluation est stratégique dans une éducation centrée sur l'apprenant, mais cela peut aussi favoriser un objectif encore plus important que l'évaluation : l'apprentissage lui-même ainsi que la responsabilité des apprentissages. Dans cette optique, la pratique réflexive et le jugement sur ses propres activités deviennent fondamentaux.

5 **Conclusion**

Cet article, après avoir exposé brièvement les principes de l'apprentissage par immersion, a mis en regard quelques artefacts d'une société de services, leur translation dans le paradigme d'immersion et les apports que cette translation apporte au dispositif pédagogique.:

- Activités d'un ingénieur du logiciel : de l'intérêt d'avoir une formation orientée métier.
- Un référentiel d'apprentissage mais aussi de production : de la multiple nature des savoirs (faire et aussi apprendre).
- Une unité de lieu, de temps et de moyens : la garantie d'une situation authentique d'apprentissage.

Toutes les procédures d'évaluation soutiennent le dispositif d'ingénierie du logiciel par immersion et facilitent le passage de l'apprentissage à la production. Les produits évalués proviennent de situations d'apprentissage authentiques, coopératives et transdisciplinaires. De plus, les connaissances et compétences sont régulées de manière récurrente et continue, et vérifiées à des moments-clés des apprentissages.

La formation est construite autour d'étapes d'apprentissage et de production, mobilisées dans les deux itérations. Cela permet :

- de prendre en compte le rôle structurant du temps et son intensité dans le monde du travail. L'apprentissage de la capacité à hiérarchiser les séquences, à maîtriser un processus global de travail sera mobilisé au-delà de cette activité précise.
- d'établir des relations coopératives entre apprenants ce qui conduit à alterner entre des temps individualisés de travail en équipe et des temps de véritable travail d'équipe.

6 **Bibliographie**

[BRJ98] Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling language User Guide, Addison-Wesley Longman, 1998

[CMMI02] CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, (CMMI-SE/SW/IPPD/SS, V1.1), <http://www.sei.cmu.edu/publications/documents/02.reports/02tr011.html>

[KET93] J.M. de Ketele, X. Rogiers, Méthodologie du recueil d'informations, Bruxelles, De Boeck, 1993

[JBR99] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addison-Wesley Longman, 1999

[RS03] Vincent Ribaud, Philippe Saliou, Software Engineering Apprenticeship by Immersion, International Workshop on Patterns in Teaching Software Development, ECOOP 2003, University of Darmstadt, Germany, 2003.

[SR07] Philippe Saliou, Vincent Ribaud, Former un praticien réflexif de l'ingénierie du logiciel, Questions de pédagogie dans l'enseignement supérieur, Université de Louvain-la-neuve, Belgique, 2007.

[Sco83] Donald Schön, The reflective practioner. New York : Basic Books, 1983.

[TIS02] TEMPO, la maîtrise du développement de systèmes informatiques, Thales Information System, 2002.

[Tar93] Jacques Tardif "L'évaluation dans le paradigme constructiviste" in L'évaluation des apprentissages. Réflexions, nouvelles tendances et formation, Sherbrooke, Université de Sherbrooke, 1993

[Tar99] Jacques Tardif, Le transfert des apprentissages Les éditions logiques, 1999.

[X50-105] Norme Afnor X50-105, Le management de projet, 1991.