

Ingénierie du logiciel par immersion

Philippe Saliou & Vincent Ribaud

EA3883, LISyC, Université de Bretagne Occidentale

29238 Brest Cedex 3, France

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

Résumé : À l'université de Brest, les étudiants en Master IUP d'Ingénierie Informatique passent les deux tiers de leur dernière année d'étude en immersion dans une 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 (celui de la société d'ingénierie Thales IS), de méthodes et d'outils associés aux architectures N-tier actuelles, mais dans des conditions d'apprentissage. Chacune des activités des processus d'ingénierie, réorientée à des buts d'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. Chaque équipe projet (groupe de cinq ou six étudiants) appelé compagnie, travaille sur sa propre plate-forme technologique (Oracle, IBM-Rational, Open Source), éventuellement avec des méthodes différentes, mais avec une même approche d'ingénierie dirigée par les modèles. Le dispositif pédagogique de la formation s'appuie sur un processus d'apprentissage intégré aux processus de production mis en œuvre dans l'industrie. Ce processus d'apprentissage se déroule en deux itérations : une première itération d'apprentissage tutoré de quatre mois permet l'acquisition « en faisant » des connaissances d'ingénierie, une deuxième itération de mise en pratique accompagnée de deux mois permet aux étudiants de mettre en œuvre (de manière autonome) les processus de production du système attendu.

Abstract : At Brest University, students studying for a Master in Software Engineering spend two-thirds of their final year immersed in a virtual company. The idea is that students follow through a project from A to Z, relying on a quality management system (used in the software engineering company Thales IS) and methods and tools associated with present n-tier architectures, but under apprenticeship conditions. Each engineering process activity, reoriented to serve apprenticeships, provides the opportunity, with the help of professional tutors, of learning appropriate knowledge and competencies in the engineering field of management information systems. Each project team (a group of 5-6 students called a company) works with its own technological environment (Oracle, IBM-Rational, Open Source), sometimes using different methods, but always with the same model-driven engineering approach. The pedagogical system of this education program relies on an apprenticeship process integrated with the production processes in force in the industry. This apprenticeship process is achieved in two iterations : during the first iteration, a 4-month tutored apprenticeship, students acquire engineering knowledge and know-how through "doing"; the second iteration, a 2-month accompanied application, allows them to put into practice (autonomously) the production processes of the expected system.

1. INTRODUCTION

Apprendre l'ingénierie du logiciel est une tâche difficile. La plupart des professionnels diront qu'ils l'ont apprise « en faisant ». Beaucoup de programmes académiques d'enseignement de l'ingénierie du logiciel s'appuient sur le paradigme d'apprentissage par projet long (1 semestre ou 1 an) en équipe [1], [2], [3]. Plusieurs universités se sont engagées dans une refonte profonde de leur curriculum d'ingénierie [4], [5].

De septembre 2002 à juin 2004, l'Université de Bretagne Occidentale a expérimenté une nouvelle forme d'éducation : l'apprentissage par immersion au sein d'un DESS d'informatique [6]. Depuis septembre 2004, cette expérimentation s'est transformée en une 2ème année de Master Professionnel dont les principaux objectifs sont la maîtrise des activités d'ingénierie du logiciel, le travail coopératif en équipe [7], la capacité à apprécier une perspective dans son ensemble et l'adaptation aux changements.

L'idée principale de ce dispositif de formation, baptisée ILI pour « Ingénierie du Logiciel par Immersion », est de faire en sorte que l'apprentissage du monde professionnel soit importé dans les murs de l'université par reproduction de ses conditions de travail : un environnement dédié, 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. En dehors des cours de langues et de communication, aucun cours traditionnel n'est dispensé. Le plan de formation est construit sur un projet de six mois en équipe, soutenu par un processus d'apprentissage spécifique, piloté et tutoré par un enseignant-chercheur issu du monde professionnel. Un véritable référentiel société (celui de Thales Information Systems) a été adapté et enrichi afin de soutenir l'apprentissage des activités d'ingénierie ainsi que la production des livrables attendus.

Dans cet article, nous énoncerons les principes de l'immersion. Ensuite, nous décrirons les spécificités pédagogiques du dispositif puis nous présenterons la manière dont nous avons intégré un processus d'apprentissage dans les processus de production mis en œuvre dans l'industrie.

2. PRINCIPES DE L'IMMERSION

L'année de formation est structurée en trois périodes (Cf. Figure 1), que l'on appelle itérations :

- Une 1^{ère} itération d'apprentissage tutoré permet l'acquisition des connaissances et savoir-faire de l'ingénierie du logiciel.
- Une 2^{ème} itération de mise en pratique accompagnée a pour objet de consolider leur transformation en compétences.
- Et, enfin, une 3^{ème} itération permet leur mise en application lors d'un stage de pré-embauche en entreprise.

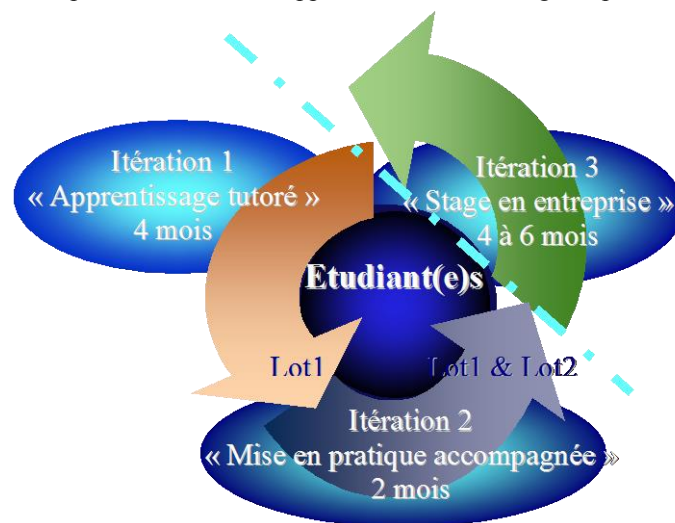


Figure 1 : Organisation temporelle

Pendant les deux premières itérations (six mois), les étudiants sont immergés dans un projet qui se déroule exactement comme en entreprise, avec les spécificités suivantes :

- *Les étudiants sont répartis en équipe de 5 appelée compagnie.* Les compagnies sont indépendantes et les étudiants doivent jouer plusieurs rôles (organisationnel, qualité, technique, soutien, etc.) dans le contexte de collaboration et de coopération lié au travail en équipe.
- *Le projet se déroule au sein de l'université, dans des salles spécialement réaménagées,* simulant l'environnement standard des entreprises de développement logiciel. Chaque compagnie dispose d'un bureau « paysager » et en commun d'une salle « machine » et de deux salles de réunion. Chaque étudiant dispose d'un poste de travail personnel.
- *Le chef de projet qui pilote et tutore le projet est un enseignant-chercheur issu du monde professionnel,* appelé tuteur de compagnie. Il s'appuie pour cela sur un référentiel d'apprentissage spécifique, construit en partie, à partir du Système de Management de la Qualité (SMQ) de la filiale informatique du groupe Thales, Thales Information Systems.
- *Une plate-forme technologique professionnelle (IBM-Rational, Oracle ou Open Source) est mise à disposition de chaque compagnie,* du fait de l'importance capitale que revêt l'architecture technique dans la construction des systèmes informatiques actuels.
- *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 Logiciel et Soutien au Développement.
- *Chaque compagnie construit un Système d'Information différent à partir d'un cahier des charges,* avec des processus similaires mais en utilisant des méthodes, des technologies et des outils différents.

Lors de la première itération d'apprentissage tutoré (quatre mois), une version incomplète du logiciel est réalisée dans un cadre totalement piloté et tutoré par le tuteur de la compagnie. Toutes les activités d'ingénierie sont alors mises en œuvre dans un cycle de développement complet.

Lors de la deuxième itération de mise en pratique accompagnée (2 mois), chaque compagnie est rendue relativement autonome afin de mettre en pratique les connaissances, compétences et savoir-faire acquis lors de la première itération. Le produit logiciel attendu à l'issue de cette itération correspond à une version complète du logiciel tel que demandé dans le cahier des charges. Lors de cette itération l'équipe reste toujours accompagnée et encadrée par le tuteur de compagnie.

La troisième itération est un stage de pré-embauche (4 à 6 mois). Il s'agit d'un stage de production dont l'objectif est la mise en application des compétences d'ingénierie du logiciel en entreprise. L'étudiant se voit alors confier la réalisation d'un petit projet de « A à Z » selon le processus de développement en vigueur dans l'entreprise ou selon le processus acquis lors de sa formation.

3. DISPOSITIF D'APPRENTISSAGE

Le dispositif d'apprentissage ILI s'appuie sur cinq éléments fondamentaux :

- Un découpage des apprentissages en domaines et activités d'ingénierie, cadre de référence commun satisfaisant à la fois aux exigences de la branche professionnelle, aux règles de l'université en tant qu'institution de formation, aux objectifs de l'équipe pédagogique ainsi qu'aux attentes des étudiants.
- Un environnement d'immersion reproduisant l'environnement d'un projet logiciel dans le domaine des Systèmes d'Information.
- Un référentiel d'apprentissage soutenant à la fois le travail des enseignants-tuteurs et les apprentissages des étudiants.
- Une redéfinition du métier d'enseignant en termes de fonctions et de rôles.
- Un système d'évaluation authentique, enchâssé dans les processus d'apprentissage-production mis en œuvre.

3.1 Domaines et activités d'ingénierie

Les activités d'ingénierie abordées lors de la formation se répartissent dans les domaines principaux de la figure 2.

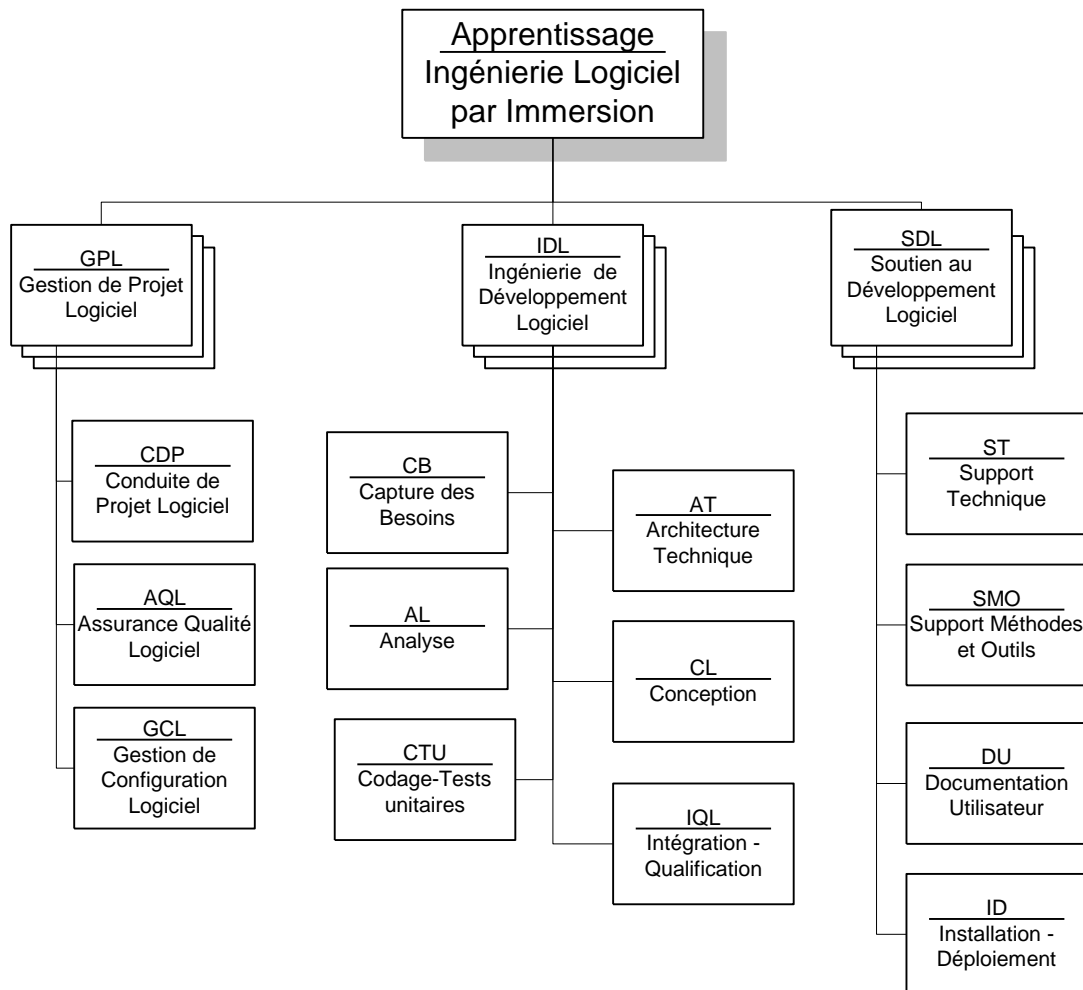


Figure 2 : Décomposition des activités

Si l'on adopte le point de vue de la branche professionnelle, il s'agit d'une décomposition du métier en trois principaux domaines d'activités (ou processus) comprenant un certain nombre d'activités. Si l'on adopte le point de vue de l'université, il s'agit de trois Unités d'Enseignement (UE) qui se décomposent en Eléments Constitutifs (EC). Si l'on prend le point de vue des étudiants et de l'équipe pédagogique, on dispose d'une nomenclature de référence de tous les apprentissages abordés.

3.2 Environnement d'immersion du projet

L'environnement d'immersion reproduit l'environnement d'un projet logiciel dans le domaine des Systèmes d'Information :

- Un espace spécialisé spécialement réaménagé comprenant un bureau « paysager » avec des postes individuels par compagnie. Les étudiants partagent également deux salles de réunion ainsi qu'une salle « machine » comprenant cinq serveurs spécifiques par compagnie.
- Un référentiel métier qui définit les pratiques et capitalise le savoir-faire du métier d'ingénieur logiciel. Grâce à un accord passé avec la société Thales-IS, nous disposons à des fins pédagogiques d'une partie de son référentiel société TEMPO [8], plus particulièrement tous les guides et procédures relatifs à l'ingénierie du logiciel.
- Un processus de développement du logiciel qui organise l'ensemble des activités permettant de transformer en produit logiciel les besoins exprimés par les utilisateurs. Le processus retenu est le 2TUP (2 Track Unified Process [9]) de la famille des processus unifiés [10].
- Une plate-forme de travail comprenant des infrastructures techniques communes et des suites d'outils destinées à la capture des exigences, à l'analyse, à la conception, à la réalisation, aux tests et à la documentation des produits logiciels.

Chaque équipe projet dispose d'une plate-forme technologique différente : Oracle, IBM-Rational ou Open Source.

Le choix de ces plates-formes s'est fait en fonction de plusieurs objectifs pédagogiques :

- Mettre à disposition des outils et des frameworks techniques en phase avec les attentes du marché, notamment le développement de systèmes d'information dans une architecture « N-tier ».
- Confronter les étudiants à la complexité des infrastructures de déploiement et de développement actuelles.
- Disposer d'outils amont de modélisation, d'outils de développement de haut niveau s'appuyant sur des frameworks techniques permettant d'avoir un développement essentiellement dirigé par les modèles.
- Disposer également d'outils bureautiques ainsi que d'outils professionnels de support au développement logiciel tels que des outils de gestion de configuration logiciel, des outils de planification, des outils de gestion des exigences, etc.

3.3 Référentiel d'apprentissage

Durant les deux premières années où la formation était expérimentale, nous avons élaboré une première version d'un référentiel d'apprentissage du métier d'ingénieur logiciel en nous appuyant sur trois choix pédagogiques essentiels : l'emploi de procédés de fabrication en phase avec les environnements actuels de développement logiciel (UML, J2EE, etc.), l'utilisation d'un référentiel ISO 9001:2000 et une évaluation enchâssée dans les situations d'apprentissages. Ce référentiel d'apprentissage soutient, d'une part, le découpage temporel en séquences d'apprentissage et, d'autre part, le processus d'apprentissage mis en œuvre lors des deux premières itérations.

Le référentiel d'apprentissage comprend à ce jour :

- Un référentiel métier TEMPO-ILI comprenant les éléments issus de l'ajustement du référentiel Thales-IS enrichis de références à d'autres référentiels de type ISO 9001 ainsi que d'exemples de documents produits sur des projets réels.
- Des supports d'utilisation des plates-formes techniques (Guides d'installation, White Papers, Red Books, etc.).
- Des ouvrages et des supports d'auto-formation.
- Des accès privilégiés de type « Academic Initiative » aux ressources d'Oracle, d'IBM et Rational.
- La description du processus d'apprentissage constitué principalement de fiches d'apprentissage génériques définissant séquence par séquence les apprentissages visés, le travail à réaliser ainsi que les fournitures pédagogiques associées.
- Les productions des promotions précédentes avec notamment l'ensemble des artefacts et documents réalisés.

Chaque activité d'ingénierie se décompose en tâches. L'apprentissage d'une tâche est décrit dans une fiche d'apprentissage dont la structure est standardisée. Elle décrit le travail à réaliser, les différentes fournitures pédagogiques sur lesquelles doit s'appuyer l'apprentissage ainsi que les produits attendus. L'étudiant perçoit la fiche d'apprentissage comme un élément essentiel d'organisation de ses compétences. Pour chaque tâche, l'élève peut associer les produits amont 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 étudiants s'appuient sur le référentiel d'apprentissage de la première itération. Ce référentiel d'apprentissage va alors être utilisé comme un référentiel de production du logiciel.

3.4 Fonctions et rôles des enseignants-tuteurs

La mise en place du dispositif d'apprentissage par immersion a nécessité une redéfinition majeure du métier d'enseignant. L'enseignant devient désormais un enseignant-tuteur auquel est affectée une fonction (ou responsabilité) principale de Tuteur-Coordinateur, Tuteur de Compagnie ou Tuteur d'Activité. Chaque enseignant-tuteur doit ensuite assumer plusieurs rôles qui représentent les différentes missions nécessaires au bon fonctionnement du dispositif : Tuteur-Client, Tuteur d'Apprentissage, etc. La notion de fonction principale est associée à la personne physique telle que perçue par l'institution et les étudiants, tandis que le rôle est équivalent à la notion d'acteur en UML. Il représente l'abstraction d'un ensemble d'activités que doit assumer un enseignant-tuteur.

3.4.1 Fonctions principales

Le « Tuteur-Coordinateur » est le responsable de la formation vis-à-vis de l'université, de l'équipe pédagogique et des étudiants. Comme pour tout responsable de formation, son travail s'effectue tout au long de l'année (organisation, gestion et suivi de la formation). Cependant, dans notre dispositif, il existe des spécificités de son travail avant le début de la formation (donner le cadre des projets, préparer la logistique, mettre à jour et mettre à disposition le référentiel d'apprentissage), pendant la première et la deuxième itération (ajuster et maintenir le référentiel, gérer les ressources, conduire et tutorer l'équipe de tuteurs), pendant la troisième itération (suivi des

stages certes mais surtout ambassadeur du dispositif auprès des entreprises) et à la fin de la formation (bilan, planification d'ajustement et préparation de la rentrée suivante).

Le « *Tuteur de Compagnie* » est l'authentique chef de projet de la compagnie. Son travail s'effectue avant le début de la formation (rédiger le cahier des charges ainsi que la proposition technique et financière), pendant la première itération (coordonner la compagnie, réguler les apprentissages, développer les compétences), pendant la deuxième itération (surveiller le bon déroulement, arbitrer et résoudre les situations de blocage) et tout au long de la formation en ce qui concerne l'évaluation.

Le « *Tuteur d'Activité* » accompagne une activité d'ingénierie. Son travail s'effectue principalement pendant la première période (formuler les objectifs de l'activité, assister la production). C'est une personne-ressource pouvant faire office de tuteur d'apprentissage sur une activité d'ingénierie particulière ou plus simplement de fournir un support à l'activité indépendamment du suivi des apprentissages.

3.4.2 Rôles

Le tableau ci-dessous répertorie les principaux rôles nécessaires au bon fonctionnement du dispositif.

Rôles	Description
Responsable de la formation	Recrutement des étudiants. Gestion des intervenants. Organisation et planification de l'année. Suivi du contrôle des connaissances. Organisation et suivi des stages. Gestion du jury. Coordination des compagnies (avancement, cohérence, indépendance, etc.).
Responsable du référentiel	Création, consolidation, adaptation et enrichissement du référentiel d'apprentissage : dispositif (principes, règles, séquençement, etc.), fiches d'apprentissage, produits et ressources pédagogiques associés. Garant du référentiel pédagogique et de son application. Relation avec Thales-IS.
Responsable logistique	Achat/Suivi matériel. Suivi/Renouvellement des contrats avec les éditeurs. Approvisionnement des outils et des licences, des ouvrages, des revues professionnelles, etc.
Tuteur de Tuteur	Accompagnement et assistance aux tuteurs de compagnie et tuteurs d'activité : régulateur, modérateur et/ou médiateur.
Tuteur-Organisateur	Définition du cadre projet spécifique à sa compagnie : besoins matériels et logiciels, choix techniques et fonctionnels, cahier des charges, proposition technique et financière.
<i>Tuteur-Chef de projet</i>	Organisation de chaque séquence d'apprentissage (ajustement, élaboration et régulations des parcours et des apprentissages individuels et collectifs). Support et assistance au quotidien de chaque étudiant. Préparation et pilotage de la réunion d'avancement hebdomadaire.
<i>Tuteur-Client</i>	Maîtrise d'ouvrage au travers d'un jeu de rôles pour les activités de capture des exigences, d'analyse, d'installation-déploiement, de recette, de maintenance (bugs, demande d'évolution).
Tuteur d'Apprentissage	Elaboration et mise à disposition du contexte et des ressources pédagogiques en lien avec l'activité abordée. Accompagnement et entraînement de chaque étudiant (Coaching). Conseil, réorientation, aide, régulation voire arbitrage en cas de blocage. Évaluation des apprentissages effectués.
Tuteur Support	Support et/ou expertise sur l'un ou l'autre des 13 domaines d'apprentissage (Qualité, Gestion de Configuration Logiciel, Support Technique, etc.) indépendamment du suivi des apprentissages.
<i>Tuteur-Modérateur</i> ^{It2}	Consolidation des produits de qualité moyenne (voire médiocre). Intervient essentiellement comme un pair sur lequel l'étudiant chef de projet ou les membres de l'équipe peuvent s'appuyer pour conforter leurs choix lors de la 2 ^{ème} itération.
<i>Tuteur-Responsable chargé d'affaires</i> ^{It2}	Responsable hiérarchique qui arbitre, régule, voire impose (respect des coûts et délais, arbitrage « hiérarchique » entre élèves). Assistance et support auprès de l'étudiant chef de projet.

3.4.3 Distribution des rôles

Un enseignant-tuteur peut jouer successivement différents rôles. Réciproquement, un même rôle peut être joué par des enseignants-tuteurs ayant une fonction principale différente. La distribution des rôles est fonction des compétences des enseignants-tuteurs ainsi que de leur expérience dans la formation. À titre d'exemple, l'arrivée d'un nouveau tuteur de compagnie obligera sans doute le tuteur coordinateur à assumer les rôles de « Tuteur-

^{It2} Rôle utile uniquement lors de la 2^{ème} itération

Organisateur » puis de « Tuteur-Client », rôles qui normalement sont à la charge des tuteurs de compagnie expérimentés.

Comme pour une formation classique, les étudiants n'ont aucune difficulté à percevoir et intégrer la fonction principale assumée par chacun des enseignants-tuteurs. Quant aux rôles, les élèves n'en perçoivent que la partie émergée (le chef de projet, le client, etc.) c'est-à-dire ceux qui sont mis en scène dans la simulation qui soutient le processus d'apprentissage. Ces rôles apparaissent en italiques dans le § 3.4.2.

3.5 Évaluation authentique

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 [11].

Lors de la 1ère période de formation « Apprentissage tutoré », la fiche d'apprentissage est le support principal de la relation Enseignants-Étudiants. Elle décrit le travail à réaliser, les différentes fournitures pédagogiques sur lesquelles doit s'appuyer l'apprentissage ainsi que les produits attendus. Chaque fiche d'apprentissage donne lieu à une évaluation formative accompagnée d'une notation. Le principe de l'évaluation formative est le suivant. Chaque fiche d'apprentissage donne lieu à un ou plusieurs produits de sortie. À l'issue d'un apprentissage, l'enseignant-tuteur évalue le(s) document(s) et produit(s) réalisés : 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. Même si le travail évalué a été réalisé seul ou en binôme, le « Tuteur d'apprentissage » effectue un débriefing en présence de toute l'équipe. Cela permet d'approfondir, de discuter, de remettre en question collégalement, les remarques apportées par le tuteur. À l'issue de ce débriefing, les étudiants concernés doivent mettre à jour et même parfois refondre entièrement leur production. Il s'ensuivra une 2ème 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 (même si dans la réalité de l'entreprise, il resterait encore beaucoup à dire). Dans ce type d'évaluation, la principale difficulté est de trouver un juste équilibre entre un jugement exigeant mais parfois sévère, plutôt guidé par le produit final auquel on est en droit de s'attendre dans la vie professionnelle, et un jugement plus modéré, plus consensuel (mais peut-être moins constructif) plutôt guidé par la dynamique individuelle et de groupe que l'on cherche à établir en situation d'apprentissage.

Lors de la deuxième itération « Mise en pratique accompagnée », les étudiants vont s'appuyer sur les fiches d'apprentissage de la 1ère période pour mettre en œuvre (de manière autonome) un processus de développement logiciel en phase avec les choix d'architecture technique, les méthodes et outils de développements utilisés ainsi que les contraintes de délais imposés par le client. Cette période doit se solder par la mise en exploitation du système qualifié conformément au protocole de validation élaboré par l'équipe projet, vérifié par le tuteur et approuvé par le client. Dans ce cas, l'évaluation s'apparente davantage à une recette ou qualification telle que pratiquée dans l'industrie. L'évaluation est divisée en différentes opérations, consistant à vérifier un ensemble de fonctionnalités, de services, de documentations ou de contraintes du système.

4. PROCESSUS 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 [12].

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, le dispositif de formation ILI 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.

4.1 Description générale du processus d'apprentissage

Le processus d'apprentissage est guidé par le cycle de développement logiciel retenu, le 2TUP (2 Track Unified Process), qui fait partie de la famille des processus unifiés.

- Il se matérialise sous la forme d'un enchaînement de séquences d'apprentissage avec des objectifs clairs et contrôlés à chaque séquence.
- À 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 ou plusieurs étudiants qui doivent alors assumer le métier (ou rôle) inhérent à l'activité concernée.
- Chaque séquence met à disposition des fournitures pédagogiques que les étudiants doivent s'approprier.
- Pour chaque fiche d'apprentissage, les étudiants 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.

On peut représenter ce processus dans un tableau à deux dimensions comme l'illustre la figure ci-dessous.

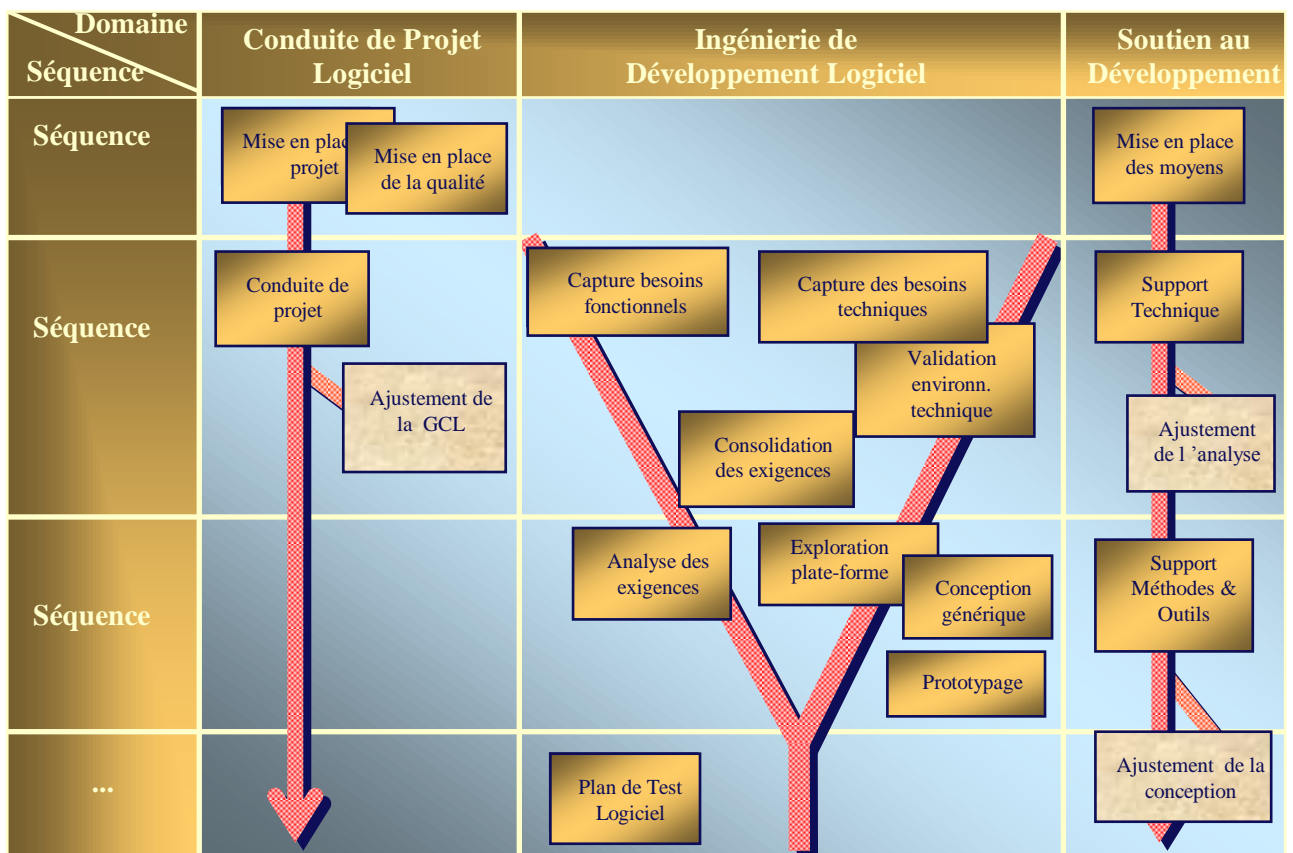


Figure 3 : Processus d'apprentissage

Les trois colonnes représentent les trois domaines principaux d'activités couverts par la formation. Ces 3 domaines peuvent également être perçus comme trois sous-processus d'apprentissage complémentaires et fortement imbriqués.

L'axe vertical de ce processus représente le temps. Il permet de matérialiser l'enchaînement des séquences et des jalons essentiels à l'apprentissage de l'ingénierie du logiciel. La durée d'une séquence peut varier de 1 à 2 (voire 3) 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 se voit confier de nouveaux travaux qui correspondent à des changements sur plusieurs axes :

- Le temps qui permet d'aborder une nouvelle activité d'ingénierie du logiciel s'appuyant sur les activités et travaux réalisés antérieurement mais qui ne prend du sens qu'à ce moment.
- Les apprentissages que nécessite la nouvelle activité et qui ne sont pas forcément dans la continuité de ses activités précédentes.
- Le domaine d'activité de rattachement : un étudiant peut passer de la conduite de projet à une action de soutien au développement, du soutien à l'ingénierie, etc.
- La spécificité « métier » de la nouvelle tâche : l'étudiant change de rôle, de chef de projet à responsable de la gestion de configuration du logiciel, d'analyste à ingénieur système, etc.

Lors de la deuxième itération, une organisation fixe du travail est mise en place. Un étudiant assume la fonction de chef de projet, les autres assurant les autres métiers inhérents à la conception et à la fabrication d'un logiciel : analyste, concepteur, développeur, etc. L'objectif est alors d'appliquer de manière autonome le processus de conception et de fabrication du logiciel acquis lors de la première itération, avec pour finalité la livraison d'une version complète du logiciel attendu.

4.2 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 [13]. 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 au transfert de connaissances et de compétences. 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 étudiants sur les aspects techniques, méthodologiques ou sur tout autre activité du processus de développement logiciel (capture des besoins, analyse, conception, etc.). Les ajustements expérimentés à ce jour s'inscrivent le plus souvent dans une démarche pédagogique inductive de type rétro-ingénierie (reverse engineering), qui revient à analyser, décomposer pas à pas une activité afin de se demander comment elle est mise en œuvre dans une situation donnée [14].

Ces activités d'ajustement apparaissent sous la forme de dents légèrement inclinées vers le bas (Cf. Figure 3), d'où le nom de processus d'apprentissage en herse par analogie à la herse agricole à étrilles (préconisée pour une agriculture écologique et durable) dont les dents flexibles peuvent être abaissées ou levées afin de préparer telle ou telle superficie à la profondeur voulue. Cette notion d'ajustement d'activité peut également être perçue comme une forme d'ajustement continu [15] très proche de la notion « d'amélioration continue » prônée par l'ISO 9001:2000 pour améliorer les processus au sein de l'entreprise à la différence qu'elle fait partie intégrante du processus d'apprentissage. Elle rejoint également des mouvements comme le *Total Quality Management for Schools* cité dans [16].

5. CONCLUSION ET PERSPECTIVES

Après trois années d'élaboration et d'expérimentation auprès de nos étudiants, notre dispositif d'apprentissage par immersion semble être aujourd'hui stabilisé ou du moins arrivé à un premier niveau de maturité. Ce dernier s'explique par une maîtrise des composants essentiels du dispositif présentés dans cet article (cf. § 3) mais surtout par la présence d'un processus d'apprentissage générique dûment décrit en termes de séquences et de fiches

d'apprentissage, avec pour chaque apprentissage, les ressources pédagogiques associées. Trois années ont été nécessaires pour mettre au point cette première version de processus d'apprentissage par immersion.

À ce jour, le bilan est très positif. Cette nouvelle forme d'enseignement innovante et passionnante satisfait pleinement à nos objectifs pédagogiques initiaux (se centrer sur les compétences à développer et les situations d'apprentissage les plus fécondes plutôt que de se contenter d'enseigner des savoirs discipline par discipline, développer une pédagogie active et coopérative fondée sur le projet et le jeu de rôles) et recueille un véritable succès auprès des étudiants et des employeurs (en 2004-2005 tous nos étudiants ont été embauchés durant leur stage de fin d'étude). Il n'en demeure pas moins qu'un certain nombre de points sont à améliorer, car notre système d'immersion reste encore fragile :

- Malgré son efficacité, il s'agit d'une pédagogie complexe à mettre en œuvre qui demande un véritable engagement de la part des enseignants-chercheurs. Ceci s'ajoute à l'engagement nécessaire pour faire une recherche de qualité.
- Certains étudiants rencontrent des difficultés voire une incapacité pour faire la transition entre leurs habitudes d'études antérieures et les exigences de notre modèle d'apprentissage coopératif où chaque membre d'une équipe projet a, à tout moment, l'apprentissage d'une compétence à faire en sous-groupe de 2 ou 3. Chaque sous-groupe détient une pièce du puzzle et une forte interdépendance lie les étudiants compte tenu des résultats collectifs auxquels ils doivent parvenir par projet.
- Les enseignants doivent disposer d'une réelle expérience industrielle en ingénierie.

À moyen terme, et ce afin de pérenniser notre dispositif d'apprentissage par immersion, nous nous sommes fixé comme objectifs :

- d'intégrer deux tuteurs de compagnie issus du monde professionnel sur des postes d'enseignant-chercheur à mi-temps,
- d'adosser notre processus d'apprentissage à un Système d'Information de suivi et de pilotage,
- d'enrichir notre référentiel d'un processus d'apprentissage orienté TMA/TRA¹,
- de refondre le contexte générique actuel d'apprentissage de certaines activités, comme par exemple la gestion de configuration logiciel,
- et enfin, d'étendre notre partenariat historique avec Thales-IS à d'autres industriels ou à d'autres universités.

6. RÉFÉRENCES

- [1] Computing Curricula, Computer Science, Volume, chapitre 10, IEEE and ACM, 2001.
- [2] Bertrand Meyer : Software Engineering in the Academy ; IEEE Computer, mai 2001.
- [3] Gilda Pour, Martin L. Griss et Michael Lutz : The Push to Make Software Engineering Respectable ; IEEE Computer, mai 2000.
- [4] G. Lachiver, D. Dalle, N. Boutin, R. Thibault, J.-M. Dirand et Coll. : Redesign of Electrical and Computer Engineering Programs at Université de Sherbrooke ; in Proceedings, Canadian Conference on Engineering Education, août 2001, Victoria.
- [5] Centre for University Teaching and Learning, <http://www.puc.aau.dk/English>, University of Aalborg.
- [6] Vincent Ribaud et Philippe Saliou : Software Engineering Apprenticeship by Immersion ; International Workshop on Patterns in Teaching Software Development, ECOOP 2003, University of Darmstadt, Allemagne, 2003.
- [7] S. Dameron : La dynamique relationnelle au sein d'équipes de conception ; Le travail humain, Vol. 65, n°4, octobre-décembre 2002, pp 339-361.
- [8] TEMPO, la maîtrise du développement de systèmes informatiques, Thales Information System, 2002.
- [9] Ivar Jacobson, Grady Booch et James Rumbaugh : The Unified Software Development Process ; Addison-Wesley Longman, 1999
- [10] Pascal Roques et Franck Vallée : UML en action ; Eyrolles, 2002
- [11] Jacques Tardif : Intégrer les nouvelles technologies de l'information – Quel cadre pédagogique ? ; ESF, 1998.
- [12] Jacques Tardif : Le transfert des apprentissages ; Les éditions logiques, 1999.
- [13] Software Process Improvement and Capability determination (SPICE), Software Process Assessment - Version 1.00, <http://www.sqi.gu.edu.au/spice/docs/baseline>, 1995.
- [14] Steven Pinker : Comment fonctionne l'esprit ; Odile Jacob, 2000.

¹ Tierce Maintenance Applicative / Tierce Recette Applicative

- [15] Philippe Saliou et Vincent Ribaud : Continuous tailoring activities in software engineering, European Software Process Improvement Conference, EuroSPI 2004, Norvège, 2004.
- [16] Mary E. Huba et Jann E. Freed : Learner-Centered Assessment on College Campuses ; Allyn and Bacon, 2000.