Mise en place d'une évaluation par compétences pour un cours d'informatique de 3ème année

Nicolas Delestre

INSA 6ème Colloque Pédagogie & Formation
16 et 17 mai 2019 – INSA Centre Val de Loire

Mise en place d'une évaluation par compétences pour un cours d'informatique de 3ème année

Nicolas Delestre[1]

[1] *INSA Rouen Normandie, BP08, 685 Avenue de l'Université, F-76801 St-Étienne-du-Rouvray, nicolas.delestre@insa-rouen.fr*

Résumé

Le cours « Algorithmique avancée et programmation C » est l'un des cours d'informatique des étudiants de 3ème année du département Architecture des Systèmes d'Information de l'INSA Rouen Normandie. Les étudiants sont évalués à l'aide de 3 notes : la 1ère, théorique (50% de la note finale), calculée à partir des notes de 2 examens sur table ; la 2ème, pratique (30%), pour un examen sur machine ; enfin la 3ème (20%) correspondant à la réalisation d'un projet par groupe de 4 à 5 étudiants.

Jusqu'à présent la notation des 2 examens sur table utilisait une évaluation descendante : pour chaque exercice d'une copie, chaque erreur rencontrée décrémentait la note maximale de l'exercice. Durant l'année universitaire 2018-2019, nous avons décidé de remplacer cette évaluation descendante par une évaluation par compétences.

L'objectif de cet article est de présenter la démarche suivie, les outils utilisés, les résultats obtenus et les évolutions envisagées pour les prochaines années.

Mots-clés : Évaluation par compétences, retours d'expériences

1 Contexte

Le cours « Algorithmique avancée et programmation C » est l'un des cours d'informatique des étudiants de 3ème année du département Architecture des Systèmes d'Information (ASI) de l'INSA Rouen Normandie (INSARN). Ce cours est constitué de 14 séances de cours magistraux (CM), 14 séances de travaux dirigés (TD) et 14 séances de travaux pratiques (TP) d'une durée d'1h30 chacune. Les séances de TD servent à étudier les algorithmes et structures de données de référence [1]. Les séances de TP quant à elles ont

1. Pour être indépendant des langages de programmation, nous utilisons à l'INSARN un pseudo code « maison » s'inspirant du langage Pascal. Ce pseudo code est utilisé dans les cours d'algorithmique de plusieurs départements.

pour objectif l'apprentissage de la programmation en langage C et des bonnes pratiques de développement. Les 6 premières séances sont des sujets de TP classiques alors que les 8 dernières servent au développement d'un projet par équipe de 4 à 5 étudiants.

Les ressources pédagogiques de ce cours sont disponibles sur la plateforme pédagogique moodle de l'INSARN [2]. Les étudiants peuvent y trouver :

- un support de cours de remise à niveau à destination des « intégrés », étudiants issus de DUT (Informatique, GEII, Réseau & Télécom, Mesures Physiques), CPGE, L2 ou L3 de Mathématiques, d'Informatique ou d'EEA ;
- un QCM d'auto-positionnement intitulé « Avez-vous les connaissances nécessaires pour suivre ce cours ? » de 29 questions que les étudiants doivent faire en début d'année ;
- les supports des 13 cours (le premier cours durant 2 séances) ;
- des QCM d'auto évaluation pour les 5 premiers cours ;
- les sujets de TD dont les corrections sont accessibles une semaine avant chaque examen papier ;
- les sujets des TP avec pour chaque TP les compétences prérequises et les compétences théoriquement acquises à l'issue de la séance ;
- le sujet et planning du projet ;
- les annales des examens théoriques avec leurs corrections et des examens pratiques ;
- quelques liens vers des ressources externes comme par exemple le très bon site de David Galles de l'université de San Francisco, présentant de manière graphique le fonctionnement des algorithmes et structures de données de référence.

Les étudiants sont évalués à l'aide de 3 notes : la 1ère, théorique (50% de la note finale), calculée à partir des notes de 2 examens sur table (partiel et final) ; la 2ème, pratique (30%), pour un examen sur machine ; enfin la 3ème (20%) correspondant à la réalisation du projet.

Jusqu'à présent la notation des 2 examens sur table utilisait une évaluation descendante : pour chaque exercice d'une copie, chaque erreur rencontrée décrémentait la note maximale de l'exercice. Outre le fait que cette méthode peut poser des problèmes de cohérence de correction entre les copies, elle ne permet pas d'identifier facilement ce que les étudiants savent (faire) ou ne savent pas (faire). Or ces dernières informations sont très utiles pour l'équipe pédagogique et pour les étudiants surtout après le 1er examen sur table qui a lieu au milieu du semestre.

Lors du dernier colloque inter INSA qui s'était déroulé à Lyon en 2017, nous avions assisté à une présentation de collègues de mathématiques de Toulouse qui avaient mis en place une évaluation par compétences. Leurs retours étaient plutôt positifs. La même année, des collègues de l'université Le Havre Normandie ont transformé un Master en informatique complet en évaluation par compétences. Leurs retours étaient aussi très positifs.

Nous avons donc décidé de remplacer notre évaluation descendante par une évaluation par compétences pour les examens sur table durant l'année universitaire 2018-2019.

2 Le référentiel de compétences

Nous avons commencé par rédiger le référentiel de compétences 6 mois avant le début du semestre. Comme l'indique la figure 1, ce référentiel est organisé hiérarchiquement avec trois niveaux : domaine de compétences, sous domaine de compétences et compétences.

La décomposition du premier niveau s'inspire du cycle en V, méthodologie de développement enseignée dans ce cours. Ainsi ce niveau est composé des domaines de compétences

2. `https://moodle.insa-rouen.fr/course/view.php?id=60&section=3`

Algorithmique avancée et programmation C
Référentiel de compétences

Domaine de compétences	Sous domaine de Compétences	Code	Compétence	À maîtriser pour le partiel
Analyse				
		AN001	Savoir désigner les choses (identifiant significatif)	X
		AN002	Savoir être précis quant aux types de données utilisés	X
		AN003	Connaître le rôle de l'analyse	X
	Analyse descendante			
		AN101	Savoir identifier les entrées et sorties d'un problème	X
		AN102	Savoir décomposer logiquement un problème	X
		AN103	Savoir généraliser un problème	X
		AN104	Savoir si un problème doit être décomposé	X
		AN105	Savoir identifier un problème naturellement récursif (directement ou indirectement)	X
	Type abstrait de données			
		AN201	Savoir identifier les dépendances d'un TAD	X
		AN202	Savoir définir des TAD générique	X
		AN203	Savoir si une opération identifiée fait parti du TAD à spécifier	X
		AN204	Savoir formaliser des opérations d'un TAD	X
		AN205	Savoir formaliser les préconditions d'une opération d'un TAD	X
		AN206	Savoir formaliser des axiomes ou savoir définir la sémantique d'une opération d'un TAD	X
	Collection			
		AN301	Savoir lister les collections usuelles	X
		AN302	Savoir formaliser sous forme de TAD une collection	X
	Graphe			
		AN401	Savoir spécifier les différents types de graphes (étiqueté et/ou valué)	
Conception préliminaire				
		CP001	Savoir ce que représente le paradigme de programmation impératif	X
		CP002	Savoir ce que représente le paradigme de programmation structuré	X
		CP003	Savoir choisir entre une fonction et une procédure	X
		CP004	Savoir concevoir une signature (préconditions incluses)	X
		CP005	Savoir choisir un passage de paramètre (E, S, E/S)	X
		CP006	Connaître le rôle de la conception préliminaire	X
Conception détaillée				
		CD001	Savoir dissocier les deux rôles du développeur : concepteur et utilisateur	X
		CD002	En tant qu'utilisateur, savoir respecter une signature	X
		CD003	Savoir utiliser le principe d'encapsulation	X
		CD004	Savoir écrire des algos avec le pseudo code utilisé à l'INSA	X
		CD005	Savoir écrire un pseudo code lisible (indentation, identifiant significatif)	X

FIGURE 1 – Extrait du référentiel de compétences

« Analyse », « Conception préliminaire », « Conception détaillée », « Développement C » et « Tests unitaires ».

Certains domaines de premier niveau sont décomposés en sous-domaines de compétences. La plupart de ces sous-domaines de compétences ont été nommés. Ce n'est pas le cas lorsque les compétences incluses dans le sous domaine sont assez génériques. Par exemple le domaine de compétences « Analyse » est constitué de 5 sous-domaines de compétences, dont le premier n'est pas nommé. Les 4 autres sous-domaines sont « Analyse descendante », « Type abstrait de données, « Collection » et « Graphe ».

Enfin chaque sous-domaine de compétences est constitué d'une ou plusieurs compétences représentées par une phrase qui commence très souvent par le verbe « Savoir » suivi d'un verbe pour identifier un savoir-faire et quelques fois par le verbe « Connaître » pour identifier un savoir. Enfin chaque compétence est identifiée par un code permettant d'y faire référence dans les différents documents.

Ce référentiel de compétences a été donné aux étudiants dès le début du semestre. Il est accessible sur le site moodle du cours sous la forme d'un fichier PDF. Il a évolué une fois au milieu du semestre (ajout de certaines compétences et évolution de la nomenclature pour les codes). Il comporte aujourd'hui 70 compétences.

Ce référentiel de compétences est global, il liste toutes les compétences que les étudiants devraient avoir acquises à l'issue du semestre. Mais comme il y a un examen théorique (le partiel) au milieu du semestre, certaines compétences sont étiquetées « À maîtriser pour le partiel ».

3 Rédactions et évaluations des examens théoriques

Lors de la rédaction des deux examens sur table, nous avons listé les compétences que nous voulions évaluer. Nous avons alors créé des exercices qui nous permettaient d'évaluer ces compétences. Entre 1 et 7 compétences étaient associées à chaque question.

Ces compétences n'étaient pas indiquées dans le sujet de l'examen, mais elles étaient indiquées dans la correction que les étudiants ont pu obtenir juste après avoir passé ce dernier. Sur ce document, comme l'indique la figure 2, pour chaque question, avant sa correction, un encadré listait les compétences évaluées en distinguant celles qui étaient principales et celles qui étaient secondaires [3].

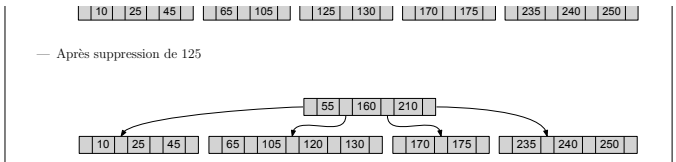
Pour chaque question, chaque compétence était évaluée à l'aide d'un échelle à 5 niveaux :

- 0 pour : la compétence n'est pas du tout acquise ou n'est pas présente sur la copie ;
- 1 pour : la compétence est utilisée mais avec beaucoup d'erreurs ;
- 2 pour : la compétence est utilisée avec quelques erreurs ;
- 3 pour : la compétence est utilisée avec peu d'erreurs ;
- 4 pour : la compétence est totalement acquise, aucune erreur.

Pour chaque question, chaque compétence a donc une « note ». Si une compétence est évaluée dans plusieurs questions, sa note est la moyenne pondérée des notes, la pondération est fonction de l'importance de la compétence dans la question (principale ou secondaire).

Enfin comme pour les jurys il faut obligatoirement remonter une note pour chaque matière, la note de la copie est la moyenne pondérée des notes des compétences évaluées, la pondération est fonction de l'importance de la compétence dans l'examen. Toutefois,

3. Ceci est facilité par le fait que les sujets d'examens sont rédigés en LaTeX, avec dans le même fichier à la fois le sujet de l'examen et la correction, et c'est dans cette partie que l'on indique quelles compétences sont évaluées pour la partie d'exercice en question. C'est à la compilation que l'on indique si on veut ou pas la correction. On obtient donc très facilement les deux fichiers PDF, sujet et correction, à l'aide d'une seule commande (*make*).



— Après suppression de 125

2 Combinaison de naturels

Après avoir expliqué votre démarche (type d'algorithme, identification de cas particulier(s), etc.), donnez l'algorithme de la fonction suivante qui calcule toutes les combinaisons possibles de listes de naturels non nuls inférieurs où égaux à une valeur n donnée :

— **fonction** combinaisons **(**n : **NaturelNonNul) :** Liste<Liste<**NaturelNonNul**>>

Par exemple l'appel *combinaisons(2)* retourne la liste $((2,1),(1,2))$ et l'appel *combinaisons(3)* retourne la liste $((3,2,1),(2,3,1),(2,1,3),(3,1,2),(1,3,2),(1,2,3))$. Votre algorithme devra respecter l'ordre de ces listes.

Solution proposée:

Sujet / question

Compétences évaluées

— Compétences principales :

CD009 Savoir écrire un algorithme qui résout le problème

CD201 Savoir identifier et résoudre le problème des cas non récursifs

CD202 Savoir identifier et résoudre le problème des cas récursifs

— Compétences secondaires :

CD004 Savoir écrire des algos avec le pseudo code utilisé à l'INSA

CD005 Savoir écrire un pseudo code lisible (indentation, identifiant significatif)

CD006 Savoir choisir la bonne itération

Compétences évaluées

Ce problème est naturellement récusif. Le cas d'arrêt est lorsque n vaut 1 et dans ce cas la valeur retournée est $((1))$. Le cas général, pour une valeur n résoud le problème pour la valeur $n-1$, on obtient donc une liste l. Et pour chaque élément de la liste l et pour chaque position possible (de 1 à $longueur(l)+1$), on insère la valeur n.

```
fonction combinaisons (n : NaturelNonNul) : Liste<Liste<NaturelNonNul>>
  Déclaration  i : NaturelNonNul
               listeDeListe, res : Liste<Liste<NaturelNonNul>>
               liste1,liste2 : Liste<NaturelNonNul>
debut
  res ← liste()
  si n=1 alors
    liste1 ← liste()
    inserer(liste1,1,1)
    inserer(res,1,liste1)
  sinon
    listeDeListe ← combinaisons(n-1)
    pour chaque liste1 de listeDeListe
      pour i ←1 à longueur(liste1)+1 faire
        liste2 ← liste1
        inserer(liste2,i,n)
        inserer(res,longueur(res)+1,liste2)
      finpour
    finpour
  finsi
  retourner res
fin
```

Correction

FIGURE 2 – Extrait de la correction de l'examen final

sur le tableau d'affichage, à l'issue des deux examens, deux documents étaient affichés : le relevé de notes et un tableau de bord.

4 Résultats pour l'année 2018-2019

Nous avons appliqué cette démarche pour les deux examens théoriques. À chaque fois 23 compétences ont été évaluées [4]. En tout, 33 compétences ont été évaluées, certaines compétences ayant été évaluées lors du partiel et lors de l'examen final. Comme indiqué précédemment, à l'issue du partiel nous avons affiché un tableau de bord pour les 23 compétences. Nous avons fait de même à l'issue de l'examen final, mais cette fois avec les 33 compétences. La figure 3 présente ce tableau de bord de synthèse. En colonne on retrouve les 33 compétences, en ligne les étudiants, ici sans leurs noms, et à l'intersection l'évaluation de la compétence. En plus de la valeur, nous avons utilisé un code couleur, plus c'est rouge plus cette évaluation tend vers 0 plus c'est vert plus elle tend vers 4. Les lignes blanches correspondent à des étudiants qui étaient inscrits au cours mais qui ne l'ont finalement pas suivi (abandon, scolarité aménagée, etc.).

Pour l'équipe pédagogique, l'évaluation par compétences a permis, après l'examen partiel, d'identifier deux groupes de compétences que la plupart des étudiants n'avaient pas bien assimilées : l'un au niveau de l'analyse, sur la décomposition d'un problème complexe, et l'autre au niveau de la conception détaillée, sur l'utilisation d'un algorithmique dichotomique dans les cas discrets. Il est à noter que ces deux groupes de compétences sont des prérequis du cours et ils ont été juste revus en début d'année, ils n'ont pas été approfondis.

Toujours pour l'équipe pédagogique, lors du jury du premier semestre, le tableau de bord de synthèse a permis de mieux comprendre les problèmes des étudiants en échec, et d'adopter des décisions que l'on espère mieux appropriées. L'avenir nous le dira...

Concernant le point de vue des étudiants, il est difficile de se prononcer. En effet, dans le rapport de l'enquête qualité sur l'évaluation des actes pédagogiques que les étudiants de l'INSARN doivent remplir à la fin de chaque semestre, l'évaluation par compétences n'a pas été mentionnée. Toutefois il est à noter que pour la première fois depuis 20 ans, aucun étudiant n'a demandé à voir sa copie, aussi bien à l'issue de l'examen partiel que de l'examen final. Pour savoir si c'était une coïncidence ou une conséquence de ce nouveau mode d'évaluation, nous leur avons envoyé un petit questionnaire avec les deux questions suivantes :

1. « Pensez-vous que l'affichage du tableau de bord d'évaluation par compétences a eu une influence sur votre besoin de consulter votre copie ? »
2. « Pour un examen futur, pensez-vous que l'affichage de votre évaluation par compétences puisse avoir une influence sur votre besoin de consulter votre copie ? »

Sur les 63 étudiants ayant suivi le cours, 33 ont répondu à cette mini enquête : 19 « oui » et 14 « non » à la première question, et 19 « oui », 13 « non » et 1 « sans opinion » à la deuxième. Même si ces résultats ne sont pas statistiquement interprétables, nous pensons que cela a eu une influence car habituellement environ un quart des étudiants demande à consulter sa copie.

5 Évolutions pour l'année 2019-2020

Nous allons donc renouveler cette expérience l'année prochaine avec les évolutions suivantes.

4. Le fait que le nombre de compétences évaluées à chaque examen soit le même est pure coïncidence.

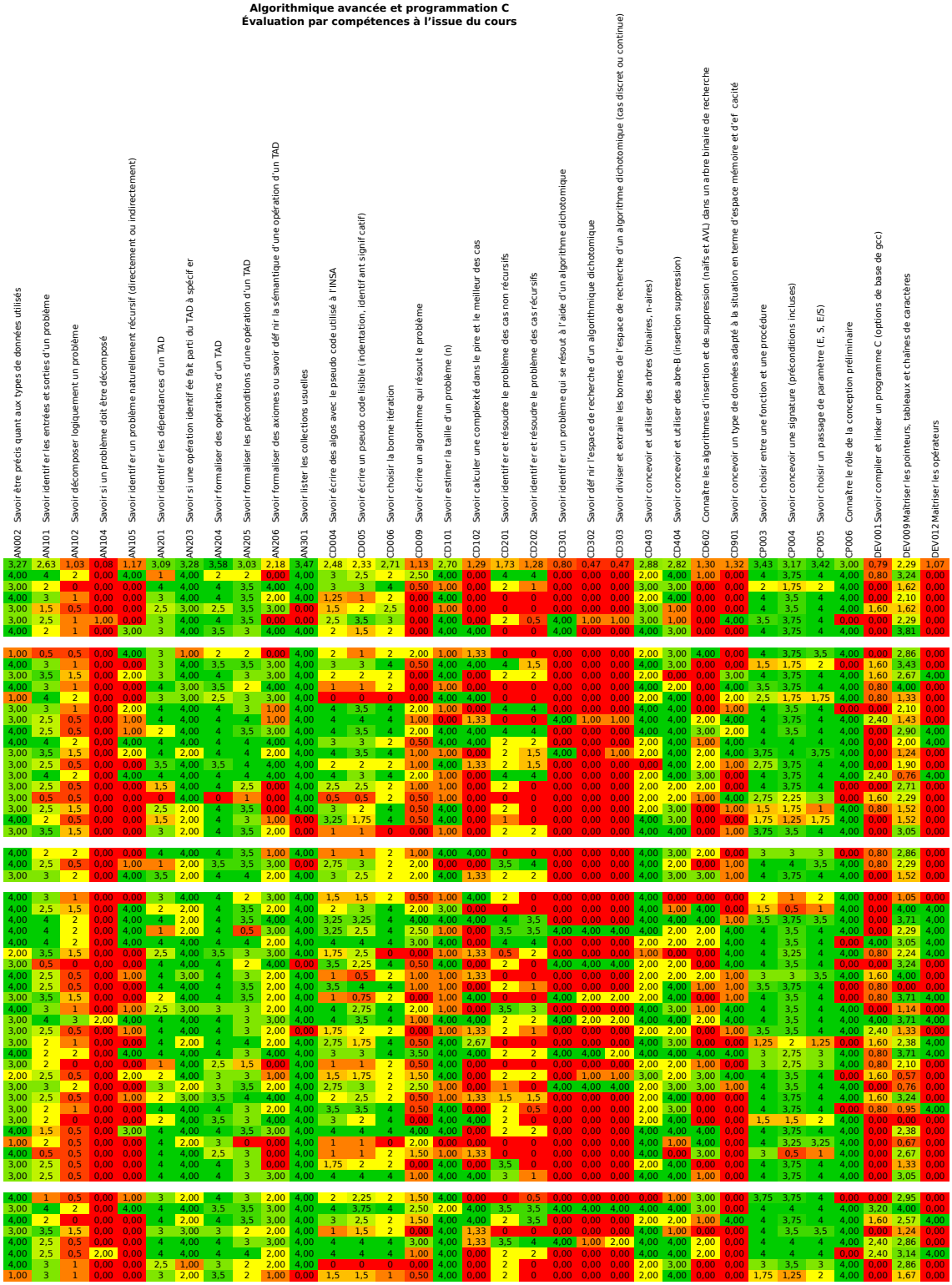


FIGURE 3 – Le tableau de bord (sans le nom des étudiants)

Tout d'abord nous allons revoir les supports de cours et le livret d'exercices de TD afin d'indiquer à chaque fois (parties de cours et exercices) quelles compétences sont enseignées ou évaluées. Nous pensons que cela permettra aux étudiants de mieux s'auto-évaluer : lorsqu'ils n'arriveront pas à faire un exercice, ils sauront quelles compétences ne sont pas acquises et quelles parties du cours ils doivent retravailler.

La deuxième évolution concerne l'examen final. Il faudrait en effet s'assurer qu'une compétence jugée importante et identifiée comme mal ou non acquise par une majorité d'étudiants à l'issue de l'examen partiel, soit de nouveau évaluée dans l'examen final. Afin que cela ne pénalise pas les étudiants, ils seront prévenus dès le début du semestre que « toute compétence considérée comme non acquise à l'issue du partiel pourra être de nouveau évaluée dans l'examen final ».

Enfin la dernière évolution, plus technique, concerne l'outil de gestion de ce référentiel de compétences et des évaluations. Actuellement nous utilisons plusieurs feuilles de tableur. Cette utilisation est très fastidieuse et source d'erreurs. Il faudrait utiliser un logiciel spécialisé. Nous avons déjà réalisé un petit état de l'art. Moodle, notre plateforme de formation, propose des outils pour gérer des référentiels de compétences et faire une évaluation par compétences, mais ils sont fonctionnellement très pauvres. Nous avons aussi identifié quelques logiciels, mais ils ont été développés pour l'enseignement primaire/secondaire (par exemple le logiciel « SACoche ») ou pour l'enseignement professionnel. Ils ne semblent pas adapter à nos besoins. Par exemple ils ne permettent pas l'injection des notes dans le système d'information de l'établissement. Ils ne permettent pas non plus la description des examens en terme questions/compétences, ni la production de code LaTeX, etc. Nous avons donc décidé d'initier le développement d'un logiciel ad'hoc.

6 Conclusion

Cet article est un retour d'expériences sur l'utilisation d'une évaluation par compétences pour un cours d'informatique de niveau L3. La mise en place de cette évaluation a été un peu chronophage en amont et pendant l'acte de formation. Cependant ceci a réellement été compensé lors de la correction des copies, lors de la remontée des notes aux étudiants et lors des jurys. Nous allons donc continuer cette expérimentation et très certainement l'étendre à d'autres cours.