Un dispositif d'évaluation continue en introduction à la programmation

Libert, Cédric; Vanhoof, Wim

Publication date:
2019

Un dispositif d'évaluation continue en introduction à la programmation

Cédric Libert[1], Wim Vanhoof[1]
[1] Université de Namur
cedric.libert@unamur.be
wim.vanhoof@unamur.be

Résumé. Dans cet article, nous décrivons un dispositif d'évaluation continue mis en place en première année de bachelier en sciences mathématiques dans le cadre d'un cours d'introduction à la programmation. Ce dispositif permet de fournir du feedback en continu aux étudiants, fait usage du portfolio et de l'évaluation par les pairs, et se base sur des parcours d'exercices qui s'inspirent d'une taxonomie des compétences adaptée à l'enseignement de la programmation. Nous présentons également les premiers résultats d'une évaluation de ce dispositif.

Mots-clés. Évaluation continue,programmation,université

1 Introduction

Dans notre université, les étudiants de première année en sciences mathématiques suivent un cours de programmation. Ce cours est classiquement divisé entre, pour moitié, des séances théoriques magistrales et, pour moitié, des séances d'exercices. Les premières ont lieu chaque semaine entre septembre et décembre, tandis que les secondes sont réparties sur toute l'année, une semaine sur deux, entre septembre et mai. L'objectif principal de ce cours est d'introduire les étudiants aux concepts de base de l'algorithmique et de la programmation impérative. La réussite de ce cours était, jusqu'à cette année, décidée sur la base d'un examen final (70 %) et de trois projets de groupes (30 %).

Depuis quelques années, deux problèmes sont soulevés concernant ce cours. D'une part, les enseignants des cours de programmation des années suivantes font part d'une maitrise insuffisante des concepts fondamentaux chez une majorité d'étudiants. D'autre part, le taux de réussite à l'examen de première session n'est pas très élevé (environ 42 %).

Pour tenter de résoudre ces problèmes, dans la lignée des travaux de De Landsheere (1971), Perrenoud (1988), Romainville (2004) et De Ketele (2010), nous agissons sur le front de l'évaluation. Cette année, un dispositif d'évaluation continue a été mis en place, dans le cadre d'un appel à projets pédagogiques innovants lancé par notre université. Ce dispositif, qui remplace l'examen final et se base sur un portfolio d'exercices rempli au fur et à mesure, sur une évaluation par les pairs et sur une observation active et individuelle de la part de l'enseignant durant les séances d'exercices. Les exercices réalisés lors de ces séances ont été adaptés de sorte à être intégrés à une taxonomie de compétences propre à l'enseignement de la programmation, inspirée de la taxonomie de Bloom.

2 Contexte, ancrages théoriques et objectifs

Comme le souligne Romainville (2004), la didactique universitaire se développe et explore différentes pistes qui permettraient de répondre à certains problèmes de la formation universitaire. Les pistes qu'il évoque pour résoudre des problèmes tels qu'observés dans notre cours de programmation – un grand taux d'échec et un problème de maitrise des acquis – consistent respectivement à donner beaucoup de feedback et à favoriser les méthodes actives. Nous mettons en place, depuis cette année, un nouveau dispositif d'évaluation en continu des apprentissages qui suit ces deux pistes. Nous présentons ici son caractère novateur au regard de trois caractéristiques de l'évaluation décrites par De Ketele (2010) : ses fonctions, ses démarches et son contenu.

De Ketele (2010) identifie trois *fonctions* possibles de l'évaluation : la fonction d'orientation, la fonction de régulation et la fonction de certification. Cette dernière a pour effet de reconnaître officiellement, devant la société, la maitrise de compétences et de connaissances. L'évaluation que nous proposions les années précédentes était majoritairement certificative. En effet, il s'agissait d'un examen final qui avait un poids important (70%) dans la réussite ou l'échec de l'étudiant. Le premier point d'action ici consiste à se centrer davantage sur la fonction de régulation, formative. En effet, Geer (2001) montre que donner une plus grande importance à cette fonction améliore l'apprentissage des étudiants. Pour ce faire, nous observons et dialoguons avec chaque étudiant lors des séances d'exercices et nous leur demandons de consigner leurs exercices dans un portfolio, accompagnés de commentaires ou de questions qu'ils se posent. En plus de l'apport métacognitif du portfolio, décrit notamment par Paulson (1990), celui-ci nous permet de suivre l'évolution d'un étudiant et de lui donner, le cas échéant, du feedback écrit. Ce suivi individuel est possible car nous somme dans une situation particulière où il n'y a généralement pas plus de 25 à 30 étudiants inscrits en première année de ce cursus. Finalement, la présentation d'exercices choisis dans le portfolio au professeur deux fois par an, en lieu et place d'un examen, contribue à la fonction certificative de l'évaluation.

Pour atteindre cette double fonction, la *démarche* utilisée est principalement de type herméneutique. Ainsi, plusieurs indices concernant l'apprentissage des concepts chez les étudiants sont rassemblés au cours de l'année, que ce soit par observation lors des séances de travaux pratiques, les discussions sur les problèmes résolus ou à résoudre, les travaux rendus en ligne, les évaluations réalisées pour les pairs et la présentation du portfolio d'évaluation. Tous ces indices servent non seulement à pratiquer de la régulation, mais sont également la base de la fonction certificative. Il s'agit, en général, d'une démarche utilisée pour évaluer des étudiants en stage. Elle est jugée pertinente par Gerard (2013), mais, il souligne qu'elle requière une certaine prudence étant donné qu'elle est principalement intuitive.

La source première des indices rassemblés est constituée des exercices que les étudiants résolvent – le *contenu* de l'évaluation. Ces exercices ont été mis en place selon une taxonomie des compétences inspirée de la taxonomie de Bloom (1952), mais adaptée aux spécificités de l'apprentissage de la programmation par Fuller (2007). Cette taxonomie, contrairement à celle de Bloom, est bidimensionnelle. Une compétence est ainsi décrite comme se trouvant à l'intersection d'un axe « interprétation » (connaitre, comprendre, analyser et évaluer) et d'un axe « production » (aucune production, appliquer, créer). Les exercices que nous avons remaniés permettent aux étudiants de suivre un chemin progressif à travers ces compétences. Une fois un exercice réussi, ils peuvent réaliser un exercice suivant d'une case directement contigüe dans la matrice. Notons que nous laissons de côté l'aspect « évaluer » de l'axe « interprétation » de la taxonomie, qui induit des compétences avancées que l'on ne cherche pas à développer dans notre cours d'introduction à la programmation.

créer		Implémenter une solution pour un problème donné	Modéliser, structurer une solution à un problème complexe en prenant en compte les cas limites
appliquer	Implémenter une solution donnée	Modifier une solution implémentée pour l'adapter à un autre problème	Débuguer : détecter et corriger des erreurs dans un programme
aucune production	Reconnaitre des éléments de vocabulaire	Vérifier / exécuter une solution à la main	
	connaitre	**comprendre**	**analyser**

Illustration 1: Taxonomie de Fuller, sans la dernière colonne "évaluation".

3. Méthodologie

Pour évaluer l'efficacité du dispositif d'évaluation, nous cherchons à estimer à quel point les étudiants maitrisent les compétences en cours d'année. La question à laquelle nous voulons répondre est la suivante : les étudiants qui profitent du nouveau dispositif mis en place ont-ils une maitrise actuelle des compétences plus grande que ceux des années précédentes ?

Pour ce faire, nous avons utilisé une séance de travaux pratiques de décembre, en milieu d'année, pour leur demander de réaliser trois exercices. Ceux-ci correspondent aux premiers exercices des examens des trois années précédentes et portent sur toute la matière qu'ils ont vue en une demi-année. Les étudiants de cette année étaient dans les mêmes conditions que ceux des années précédentes qui devaient répondre à ces questions : ils ne pouvaient pas utiliser leurs cours ni les ordinateurs, ne pouvaient pas communiquer entre eux et avaient le droit de poser des questions à l'enseignant. Deux différences majeures étaient là toutefois : ils n'ont pas été prévenus à l'avance (et donc n'ont pas « revu » le cours) et ils n'ont encore fait qu'une demi-année de programmation. Au-delà de ces deux différences, il existe évidemment un certain nombre de biais difficiles à éviter : ce ne sont pas les mêmes étudiants, le contexte n'est pas exactement le même...et donc les résultats pourraient ne pas être uniquement provoqués par le changement de dispositif.

Pour chacune des questions qu'ils avaient à résoudre, nous avons établi une liste de critères sur lesquels elles seraient évaluées. Pour chaque copie de cette année et des trois années précédentes, nous avons donné une note ternaire (0-1-2) par critère. Nous disposons donc d'une grille qui nous permet de comparer, critère par critère, nos étudiants actuels avec les étudiants des trois années précédentes.

4 Résultats et discussion

Les résultats obtenus par les étudiants de cette année et des années précédentes, sur les 26 critères choisis, ne sont pas significativement différents, en moyenne, de ceux des années précédentes. Alors qu'ils n'avaient pas préparé spécifiquement ce test et qu'ils n'ont qu'une demi-année d'entrainement. Ce dernier point est important pour un cours de programmation, car les exercices proposés sont incrémentaux, c'est-à-dire qu'ils réutilisent sans cesse les concepts vus avant et en intègrent de nouveaux à chaque séance. Ceci mène les étudiants qui les résolvent à retravailler en permanence tous les concepts acquis. On aurait ainsi pu s'attendre légitimement à ce que les étudiants des années précédentes aient des résultats supérieurs aux étudiants de cette année, puisqu'ils avaient suivi le cours complet et étaient prévenus à l'avance de l'organisation de l'examen.

Le tableau ci-dessous présente les moyennes, notées sur deux, pour un sous-ensemble des critères choisis. Ceux-ci sont représentatifs des différentes catégories de critères : éléments syntaxiques et stylistiques (indentation), concepts de programmation (déclaration de variable, utilisation de l'opérateur modulo, utilisation de structures conditionnelles if-else, de boucles), et algorithmique (calcul de moyenne et calcul de maximum).

		indentation	déclaration	modulo	If-else	boucle	moyenne	maximum
actuels	Moyenne	1,52	1,64	1,72	1,08	1,28	0,68	0,32
	Écart-type	0,81	0,74	0,66	0,93	0,83	0,84	0,68
précédents	Moyenne	1,62	1,76	1,6	1,07	1,41	0,31	0,41
	Écart-type	0,72	0,62	0,8	0,91	0,89	0,53	0,72

En toute prudence, on pourrait conclure que, au pire, le dispositif mis en place n'a pas eu d'impact négatif sur la maitrise des concepts. On pourrait également s'avancer un peu plus en affirmant que, si les résultats sont comparables à la moitié de l'année, il est possible qu'après une demi-année d'entrainement supplémentaire les étudiants actuels obtiendront de meilleurs résultats. Un nouveau test sera ainsi organisé en mai pour vérifier si cette tendance se confirme.

5 Conclusion

En nous inscrivant dans une démarche d'évaluation continue pour évaluer cours de programmation, nous avons proposé un dispositif qui s'inscrit dans la lignée des travaux de Romainville (2004) et De Ketele (2010) et repose sur trois leviers.

Le premier est une volonté de donner beaucoup de feedback personnalisé à chaque étudiant lors des séances d'exercices. Cela n'est possible évidemment, ici, que parce que le nombre d'étudiants n'est pas trop élevé (27 inscrits). Cette importance donnée au feedback devrait permettre de garantir une meilleure maitrise des compétences.

Le second est la réalisation, par chaque étudiant, d'un portfolio où ils consignent des exercices et des remarques, d'eux-mêmes ou de l'enseignant. Le portfolio a un avantage métacognitif, permet de suivre l'évolution de l'étudiant, et participe à la fonction certificative de l'évaluation, par la présentation d'exercices du portfolio lors d'une entrevue semestrielle avec l'équipe enseignante.

Le troisième est un remaniement des exercices au regard d'une taxonomie de compétences prévue pour les cours de programmation, et la possibilité de suivre plusieurs chemins d'apprentissages parmi ces exercices.

L'évaluation de mi-année qui a été réalisée est encourageante dans la mesure où les étudiants de cette année ne semblent pas avoir plus de difficultés que les étudiants des années précédentes à résoudre, par surprise, des problèmes posés à des examens. Une seconde évaluation aura lieu au mois de mai, ainsi qu'un questionnaire de satisfaction à destination des étudiants. Ceci clôturera la première itération de ce dispositif, qui évoluera en fonction des résultats obtenus.

Références

Bloom, B. S. (1956). Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay*, 20-24.

Cuban, L. (1986). *Teachers and machines, the classroom use of technology since 1920*. New York and London: Teachers College Press.

De Ketele, J. (2010). Ne pas se tromper d'évaluation. *Revue française de linguistique appliquée*, vol. xv,(1), 25-37. https://www.cairn.info/revue-francaise-de-linguistique-appliquee-2010-1-page-25.htm.

De Landsheere, G. (1971). Evaluation continue et examens: précis de docimologie.

Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J. & Thompson, E. (2007, December). Developing a computer science-specific learning taxonomy. In *ACM SIGCSE Bulletin* (Vol. 39, No. 4, pp. 152-170). ACM.

Gerard, F.-M. (2013), L'évaluation, un levier pour la réussite, *Après l'université d'été... la feuille d'automne - Actes de l'Université d'été des enseignants de la CCI Paris Ile-de-France*, 27-28 juin 2013, 1-7.

Greer, L. (2001). Does changing the method of assessment of a module improve the performance of a student?. *Assessment & Evaluation in Higher Education*, *26*(2), 127-138.

Linn, M. C. (2003). Technology and science education : starting points, research programs and trends. *International Journal of Science Education, 25*(6), 727-758.

Morin, E. (1990). *Introduction à la pensée complexe*. Paris: Le Seuil.

Perrenoud, P. (1988). La part d'évaluation formative dans toute évaluation continue.

Paulson, L. F., & Paulson, P. R. (1990). How Do Portfolios Measure Up? A Cognitive Model for Assessing Portfolios.

Romainville, M. (2004). Esquisse d'une didactique universitaire. *Revue francophone de gestion*, *5*, 24.