Développement dirigé par les tests et revue de code par les pairs pour l'apprentissage de la programmation

Franck Silvestre, Jean-Baptiste Raclet

Développement dirigé par les tests et revue de code par les pairs pour l'apprentissage de la programmation

Franck Silvestre, Jean-Baptiste Raclet
Université Toulouse 3, Institut de Recherche en Informatique de Toulouse
franck.silvestre@irit.fr
jean-baptiste-raclet@irit.fr

Résumé. Cet article introduit une nouvelle approche d'enseignement de la programmation s'inspirant de la démarche d'apprentissage dirigé par les tests (TDL). Cette approche, baptisée "apprentissage dirigé par les tests et la revue par les pairs" (PRaTDL), est caractérisée par un protocole articulant les séquences de programmation en autonomie dirigée par les tests et les séquences de revue de code par les pairs. La démarche PRaTDL a été expérimentée sur un groupe d'étudiants de Master Informatique. Les résultats de notre étude qualitative mettent en avant les sentiments d'autonomie et d'encapacitation associés aux séquences de programmation dirigée par les tests et les sentiments d'approfondissement et de compréhension suscités par les séquences de revue de code par les pairs.

Mots-clés. Apprentissage de la programmation, développement dirigé par les tests, revue de code par les pairs, évaluation par les pairs.

1 Introduction

L'apprentissage dirigé par les tests (TDL pour *Test-Driven Learning*) a été introduite par Janzen & Saiedian (2006) pour relever le défi d'apprendre à programmer en utilisant les tests.

Notre travail s'inscrit dans la continuité des travaux de Janzen & Saiedian (2006, 2008) pour l'apprentissage de la programmation et plus précisément sur l'acquisition de capacités à discuter des solutions mises en œuvre dans la résolution d'exercices ou de problèmes. Nous décrivons dans un premier temps le contexte de nos travaux, leurs ancrages théoriques et les objectifs poursuivis. La deuxième section décrit le protocole mis en œuvre et la méthodologie déployée pour recueillir les bénéfices perçus par les étudiants sur le protocole proposé. Enfin, nous présentons les résultats de notre expérimentation menée sur un groupe d'étudiants inscrit dans un cursus de Master 2 Informatique en développement logiciel.

2 Contexte, ancrages théoriques et objectifs

En ingénierie logicielle, les tests représentent un dispositif incontournable pour vérifier et valider un système logiciel (Bourque & Fairley, 2014). L'adoption massive des méthodes agiles dans l'industrie (Stavru, 2014) a mis de facto sur le devant de la scène le développement dirigé par les tests (TDD pour *Test Driven Development*). Cette pratique d'ingénierie, formalisée initialement dans le cadre de la méthode XP (Beck, 2003), consiste à développer les fonctionnalités d'un logiciel dans une succession de cycles commençant par l'écriture d'un test, suivi de l'écriture du code faisant passer le test et terminant par l'amélioration du code principal (*refactoring*). Cette approche redéfinit complètement le rôle des tests en ingénierie logicielle : le TDD est une méthode d'analyse, de conception et de développement reposant sur l'écriture de tests automatisés tout au long du cycle de développement (Beck). L'usage des tests est, dans ce cadre d'utilisation, pervasif, et ne se cantonne plus à un seul objectif de vérification du système logiciel en aval du développement réalisé. Pourtant le TDD ne suit pas la courbe d'adoption des méthodes agiles (Causevic, Sundmark, & Punnekkat, 2011). Dans leur étude, Causevic, Sundmark, & Punnekkat énoncent sept facteurs de limitations de l'adoption du TDD dans l'industrie. Parmi les sept, deux sont directement liés au manque de connaissances et de compétences sur le TDD ainsi que sur l'écriture des tests.

L'apprentissage dirigé par les tests (TDL pour *Test Driven Learning*) a été introduit par Janzen & Saiedian (2006, 2008) pour relever le défi d'apprendre à programmer en utilisant les tests pour expliciter l'usage et le comportement de portions de codes dans différents langages. Les résultats des expérimentations menées par Janzen & Saiedian (2006) montrent que les étudiants gagnent en compréhension des concepts lorsqu'ils sont en situation d'apprentissage dirigé par les tests. Nos travaux s'inscrivent dans la continuité des travaux amorcés par Janzen & Saiedian (2006, 2008). Nous avons proposé à nos étudiants un protocole visant à compléter une

approche uniquement dirigée par les tests par des activités de revue de code. Nous avons souhaité ainsi cibler un ensemble de résultats d'apprentissages plus large concernant principalement les capacités à discuter sur différentes solutions apportées à un problème de programmation (ACM, 2013). Pour cet article, nous nous sommes intéressés aux bénéfices perçus par les étudiants lors de la mise en œuvre de cette nouvelle démarche.

3 Méthodologie

Notre travail s'inscrit dans une démarche de recherche-action visant l'amélioration de l'apprentissage de la programmation. L'expérimentation a été menée dans une unité d'enseignement de Master Informatique portant sur l'apprentissage d'interfaces de programmation d'applications (API) en Java.

Dans cette unité d'enseignement, le protocole a été le suivant :

- Les étudiants ont à développer individuellement un projet dont les fonctionnalités peuvent être conçues à l'aide de l'API visée.
- Les consignes générales sont indiquées sur un format standard d'énoncé rédigé en langage naturel.
- Les questions exigeant en réponse l'écriture de code sont systématiquement accompagnées du jeu de tests automatisés devant passer avec succès pour considérer que l'exercice a été accompli correctement.
- Chaque séance est structurée sous la forme d'une alternance de séquences de travail en autonomie sur le projet (phases « solo » pendant lesquelles l'apprentissage est dirigé par les tests) et de séquences de revues par les pairs sur les solutions mises en œuvre (phases « duo »).

Nous définissons la notion d'« Apprentissage du développement dirigé par les tests et la revue par les pairs » (PRaTDL pour *Peer Review and Test-driven Development Learning*) comme démarche d'apprentissage de la programmation supportée par le protocole proposé ci-dessus.

Les séquences de revue par les pairs ont été mises en œuvre en utilisant la plateforme Tsaap-Notes (Silvestre, Vidal & Broisin, 2017). Celle-ci permet à l'enseignant de poser des questions aux étudiants et d'orchestrer un processus en 3 étapes : (1) pour chaque question, chaque étudiant, à l'aide d'un dispositif connecté, fournit une réponse au système ; (2) le système demande à chaque apprenant d'étudier et d'évaluer jusqu'à cinq contributions apportées par les autres étudiants ; (3) après publication des résultats, l'enseignant et les étudiants ont accès à la liste de toutes les contributions ordonnées de la mieux notée à la moins bien notée. L'enseignant et les étudiants ont alors l'opportunité d'échanger sur les résultats observés.

Afin de mesurer les bénéfices perçus par les étudiants quant à l'approche PRaTDL, nous avons proposé aux étudiants de Master un questionnaire détaillé dans le tableau 1.

Tableau 1. Questionnaire soumis aux étudiants

N°	Question	Domaine de valeurs des réponses
1.1	J'ai appris beaucoup durant les 8 premières séances de cette UE. Sur une échelle de 1 à 10 jusqu'à quel point êtes-vous d'accord ou pas d'accord avec cette affirmation ?	Echelle linéaire de 1 à 10. 1 = Pas du tout d'accord ; 10 = Tout à fait d'accord
1.2	Justifiez votre réponse.	Texte libre.
2.1	Je tire complètement partie de la phase "solo".	Echelle linéaire de 1 à 10. 1 = Pas du tout d'accord ; 10 = Tout à fait d'accord
2.2	Que m'apportent les séquences "solo" guidées par les tests ?	Texte libre.
3.1	Je tire complètement partie de la phase "duo".	Echelle linéaire de 1 à 10. 1 = Pas du tout d'accord ; 10 = Tout à fait d'accord
3.2	Que m'apportent les séquences "duo" pour répondre aux questions de compréhension ?	Texte libre.

4 Résultats et discussion

Sur les 24 apprenants ayant participé à l'expérimentation, 22 apprenants ont répondu au questionnaire présenté dans le tableau 1. De leurs réponses émerge un fort sentiment d'apprendre. En effet, à la question 1.1, la totalité des étudiants répond avec une note supérieure à 5 ; ils sont 14 (64%) à y répondre avec une note supérieure à 7 sur 10. Ce résultat traduit la perception par les étudiants d'un approfondissement dans les apprentissages sachant que cet enseignement est de niveau avancé et s'inscrit dans la continuité directe de plusieurs enseignements reçus en Master 1. De plus, il concerne une API que certains apprenants avaient mis en œuvre au cours de leur stage de Master 1.

Concernant les modalités d'apprentissage, la phase de travail individuel s'appuyant sur le TDD est plébiscitée par les apprenants. Au moment d'évaluer l'affirmation proposée à la question 2.1, ils sont 21 (95.5%) à y répondre avec une note supérieure à 5 et 16 (72.7%) avec une note supérieure à 7. Le nuage de mots de la Figure

1.a synthétise les termes les plus fréquents dans les justifications fournies en réponse à la question 2.2; il fait apparaître un sentiment d'émancipation (« autonomie », « découvrir », « seul », « recherche »), d'encapacitation (« pouvoir », « permet ») et d'apprentissage (« savoir », « apprendre »).
Enfin, la question 2.2 a permis d'évaluer la perception des étudiants sur la phase de confrontations et d'échanges. 21 (95,5%) apprenants répondent avec une note supérieure à 5 et 19 (86.4%) avec une note supérieure à 7. Le nuage de mots de la Figure 1.b regroupe les mots les plus fréquents des commentaires des étudiants sur cette phase. Ces séquences font naître un sentiment d'approfondissement d'une part (« mieux », « réflexion », « confronter », « constructif ») et de compréhension d'autre part (« comprendre », « connaissance », « effectuer »).

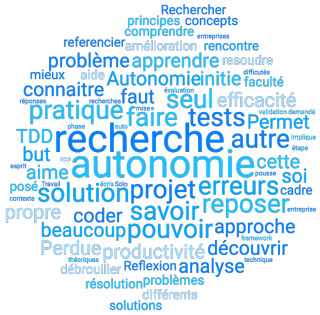

Figure 1.a Nuage de mots extraits des réponses aux questionnaires remplis par les apprenants pour la phase "solo"

Figure 1.b Nuage de mots extraits des réponses aux questionnaires remplis par les apprenants pour la phase "duo"

5 Conclusion

Cet article introduit une nouvelle approche d'enseignement de la programmation s'inspirant de la démarche d'apprentissage dirigée par les tests (TDL). Cette nouvelle approche, baptisée « apprentissage dirigé par les tests et la revue par les pairs » (PRaTDL), est caractérisée par un protocole articulant les séquences de programmation en autonomie dirigée par les tests et les séquences de revue de code par les pairs afin de couvrir l'essentiel des résultats d'apprentissage décrits dans le référentiel ACM (2013).

Nous avons expérimenté l'approche PRaTDL sur un groupe d'étudiants de Master en Informatique. Les séquences de revue de code ont été, pour cette expérimentation, orchestrées par la plateforme en ligne Tsaap-Notes. Nous avons mesuré qualitativement les bénéfices perçus par les étudiants sur la mise en œuvre de l'approche PRaTDL. Les résultats de l'étude qualitative mettent en avant les sentiments d'autonomie et d'encapacitation associés aux séquences de programmation dirigée par les tests et les sentiments d'approfondissement et de compréhension suscités par les séquences de revue de code par les pairs.

Références

ACM, Force ACM Joint Task (2013). *Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science*. Technical report, Association for Computing Machinery (ACM) IEEE Computer Society.

Beck, K. (2003). *Test-driven development: by example*. Addison-Wesley Professional.

Bourque, P., & Fairley, R. E. (2014). *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press.

Causevic, A., Sundmark, D., & Punnekkat, S. (2011, March). Factors limiting industrial adoption of test driven development: A systematic review. In *Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth International Conference on*(pp. 337-346). IEEE.

Janzen, D. S., & Saiedian, H. (2006, March). Test-driven learning: intrinsic integration of testing into the CS/SE curriculum. In *ACM SIGCSE Bulletin* (Vol. 38, No. 1, pp. 254-258). ACM.

Janzen, D., & Saiedian, H. (2008, March). Test-driven learning in early programming courses. In *ACM SIGCSE Bulletin* (Vol. 40, No. 1, pp. 532-536). ACM.

Silvestre, F., Vidal, P., & Broisin, J. (2017). Un nouveau processus d'évaluation pour améliorer la qualité des feedbacks dans les tests en ligne. *Sticef, 24(1)*.

Stavru, S. (2014). A critical examination of recent industrial surveys on agile method usage. *Journal of Systems and Software*, *94*, 87-97.