



HAL
open science

Dispositif hybride pour développer l'expertise numérique des étudiants ingénieurs non développeurs: Apprendre collaborativement en faisant pour être apte à concevoir des cahiers des charges et réceptionner des applications

Walter Nuninger

► To cite this version:

Walter Nuninger. Dispositif hybride pour développer l'expertise numérique des étudiants ingénieurs non développeurs: Apprendre collaborativement en faisant pour être apte à concevoir des cahiers des charges et réceptionner des applications. 7ème colloque "Transformations et numérique: quels défis à relever et quelles responsabilités à prendre?", AUPTIC.Education, Nov 2023, Caen (FR), France. hal-04334760

HAL Id: hal-04334760

<https://hal.science/hal-04334760>

Submitted on 11 Dec 2023

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.

Copyright

1. Titre de la communication indépendante

Dispositif hybride pour développer l'expertise numérique des étudiants ingénieurs non développeurs.

2. Auteur(s)

1 ^{er} auteur	Walter NUNINGER Walter.Nuninger@univ-lille.fr Univ. Lille, CNRS, Centrale Lille, URM 9189 CRISTAL, F-59000 Lille, France
2 ^{ème} auteur	
3 ^{ème} auteur	
4 ^{ème} auteur	

(ajoutez des lignes s'il y a plus de 4 auteurs)

3. Types (soulignez ce qui convient)

<u>Analyse de pratiques</u>	<u>Développement d'outils</u>
Travaux de recherche	Débat et point de vue

4. Axe thématique (soulignez ce qui convient)

Intelligence artificielle	Genre
Immersivité	<u>Vers une responsabilité partagée de l'accompagnement</u>

5. Résumé court type teaser (250 mots max. avec une brève bibliographie indicative)

L'usage professionnel du numérique de nos élèves ingénieurs non informaticiens relève du dernier niveau du référentiel DIGCOMP 2.2 (Vuorikari et al., 2022). Il s'agit, à partir des instructions communes des langages de programmation, de développer les aptitudes nécessaires à l'interaction avec les développeurs dans le cadre de projets informatiques, collaboratifs et agiles (Printz, 2010). Basées sur l'analyse du besoin, l'objectif est la conception de solutions innovantes par intégration de fonctionnalités existantes ou nouvelles, avec la prise en compte des limites et le souci de performance et de capitalisation (clean code, maintenance/évolution (Martin, 2009)). Pour développer cette *pensée informatique*, les attendus pédagogiques priorisent l'analyse fonctionnelle descendante (IDEF0) et les tests unitaires d'échec (Martraire et al., 2022) en exploitant le langage naturel (pseudo-code). La pédagogie est active avec un cours inversé et hybride qui inclut une autoformation (Moodle synchronisé, Scratch). L'environnement de programmation Scilab (logiciel libre de calcul numérique et de simulation) permet une mise en situation formative (pair programming, code review) accolée à un processus progressif d'évaluation par les pairs (Thomas et al., 2011). Le TP-projet contribue à l'intelligence collective et à une dextérité transférable. Cependant, le manque d'autonomie observé et le faible engagement sur la durée limitent la levée des freins pour les publics sans

expérience de programmation (séquence d'appel et affectation des variables typées) mais aussi, pour les adeptes du codage, au regard du changement de paradigme (découpage fonctionnels et tests). Certains leviers relèvent des parcours antérieurs et de l'organisation contrainte qui limite la souplesse requise de réalisation.

6. Résumé long (1000 mots max)

Pour les élèves ingénieurs non informaticiens, dès le début du parcours de trois ans, on vise à développer un usage professionnel du numérique. A l'issue du module « informatique » (36h) on vise le niveau « highly specialised » du référentiel DIGCOMP 2.2 (Vuorikari et al., 2022) pour une « utilisation confiante, critique et créative des TIC pour atteindre les objectifs liés au travail, à l'employabilité, à l'apprentissage, aux loisirs, à l'inclusion et / ou à la participation à la société » ; compétences clés d'une formation tout au long de la vie (EU, 2019). L'objectif est l'aptitude à interagir avec les développeurs dans le cadre de projets informatiques, collaboratifs, agiles et bien conçus (Martin, 2009 ; Printz, 2010) qui répondent aux besoins spécifiques et innovent en suivant l'évolution des outils et méthodes. Le scénario pédagogique soutient la résolution de problème et la production numérique, la collaboration et la sécurité, tout autant que la culture de la donnée et de l'information. Dans le cadre de la filière Génie Biologique et Alimentaire, le public est hétérogène en parcours (16% DUT, 20% L2, 33% de préparation intégrée, 31% classes préparatoires) et féminin à 80% (46 étudiant·es). La pédagogie retenue est active avec un cours hybride qui tend vers la classe inversée et intègre des mises en situations pour « apprendre en faisant » (Grzega, 2005). Le cours débute avec un mini-jeu pédagogique et sérieux pour créer la dynamique de groupe et découvrir l'analyse fonctionnelle descendante. Il est précédé d'une autoformation avec ONAAG (Outil Numérique d'Auto Apprentissage Guidée). Ce Moodle synchronisé permet d'amorcer l'autonomie d'apprentissage et l'usage numérique en tant qu'utilisateur, d'identifier les attendus, de sensibiliser à l'approche pédagogique et d'engager dans le parcours avec les pairs (Nuninger, 2017). En absence de prérequis, une ressource Scratch est incluse pour faire comprendre à l'aide de la programmation par blocs (Code.org, 2022) la construction des traitements pour le but (algorithmes). Une série de C/TD vise l'appropriation des cinq catégories d'instructions communes aux langages de programmation : affectation (variables et formats), lecture/écriture, alternatives (tests), boucles (itérations) et les listes (structures). Le TP-Projet (50% du temps de l'UE, en sous-groupe de 6 étudiants) s'appuie sur l'environnement de programmation Scilab pour créer une application « originale ». Ce logiciel open source, utile à l'ingénieur, est un IDE (Integrated Development Environment) dédié au calcul numérique et à la simulation dont le lexique est riche et les syntaxes habituelles au regard des principaux langages (C++, Java, Python, R...). L'intérêt est une dextérité nouvelle transposable tout en obligeant de sortir des habitudes éventuelles (Python). Pour développer la *pensée informatique*, les attendus pédagogiques priorisent l'analyse fonctionnelle descendante (IDEF0) et les tests unitaires d'échec (Martraire et al., 2022) en exploitant le Langage de Description d'Algorithme (LDA : langage naturel, pseudo-code). A partir de la définition du besoin (brainstorming, cahier des charges), l'effort relève de la conception innovante par intégration de fonctionnalités existantes ou à développer mais validées, et ainsi proposer des services supplémentaires. Les traitements sont décrits de manière universelle, moins dépendants de la syntaxe d'un langage spécifique ou des choix séquentiels, avant le passage à la réalisation opérationnelle pour donner du sens. Plus technique, le codage des algorithmes s'en trouve facilité. Le groupe doit alors prendre en compte les limites des propositions (correction et vérification des données d'entrées des séquences d'appels) dans un souci de performance et de maintenance (ré-usinage pour un code minimal et « clean »). Au TP-projet formatif, est accolé un processus progressif SCPR d'évaluation progressive et partagée (Nuninger et al., 2022) qui doit soutenir l'évolution

individuelle et collective tout en sensibilisant aux valeurs et principes agiles dont la méthode Scrum (Radenkovic, 2012) : prioriser la valeur ajoutée pour le client et recalculer du fait des évolutions. Le cadrage du projet impose, sur base des expertises individuelles, une production par lots des fonctions (pair programming en trinôme) validée de manière croisée (code review entre pairs du groupe de TP) et qui sont ensuite exploitées par sous-groupe (mob programming) pour créer des applications concurrentes ; objets de l'évaluation par les pairs (Thomas et al., 2011 ; Roy & Michaud, 2018). En contribuant à l'intelligence collective le TP-Projet crée une opportunité individuelle mais en 2023, contrairement aux années antérieures et de manière plus générale, on observe un manque d'autonomie (besoin permanent d'assistance sans en tirer bénéfice) et un faible engagement sur la durée (recherche d'information, résolution de problème...). Cela limite la levée des freins habituels pour les publics sans expérience de programmation (séquence d'appel et affectation des variables typées) mais aussi, pour les adeptes du codage, au regard du changement de paradigme (découpage fonctionnels et tests). Les causes semblent générationnelles, liées au processus de recrutement face à une diversité des parcours de publics entrés à l'université pendant ou à l'issue de la crise sanitaire, mais aussi à la constitution laissée libre des sous-groupes. On constate une moindre maîtrise de la langue avec des difficultés à appliquer des standards mais aussi à échanger dans un collectif et à capitaliser. Certaines postures limitent alors la trajectoire habituelle d'entraînement du groupe. Les attendus sont évalués par une épreuve finale individuelle couvrant les 3 niveaux d'expertise : surface, intermédiaire et profondeur. Pour les étudiantes (80% du groupe), les médianes sont similaires sauf pour les DUT tandis que pour les étudiants les écarts ne sont pas explicables du fait du faible effectif. Les résultats montrent une variabilité liée aux parcours initiaux et à la constitution des groupes TD ce qui confirme les observations faites en séances par les intervenants respectifs. Par comparaison, l'évaluation finale de la promotion 2022 (déjà entrée à l'université en 2019 avec une composition semblable en parcours) respecte une répartition normale sans différence par groupe de TD ou par genre au niveau de la promotion (médiane). Les enquêtes de début et fin de module permettraient l'analyse des motivations initiales et ressenties finales par groupe au regard de la note mais le nombre de répondants est insuffisant (<40% en 2023). Les réponses traduisent néanmoins des croyances, incompréhension et désabusement, voire rancœur. Le taux de répondants jugeant avoir progressés baisse de 10 points et la satisfaction globale de 20 points en 2023 ce que ne saurait expliquer la seule introduction des tests unitaires. Certains leviers d'accompagnement personnalisés et de mise en confiance n'opèrent pas du fait des modalités opérationnelles (maquette, emploi du temps, effectifs) et malgré la souplesse du scénario. Il conviendrait, en amont, de renforcer l'aisance numérique et la capacité d'apprentissage (organisation, gestion du temps) et le collaboratif plus que la coopération pour mieux orienter vers le parcours ingénieur.

7. 3-5 mots-clés

Conception fonctionnelle, Test Driven Développement (test unitaire d'échec), Clean code, Agilité, LMS synchronisé, Evaluation par les pairs, Apprendre en faisant, Collectif

8. Bibliographie indicative

Bibliographie du résumé court.

Martin, C. (2009) *Coder proprement*, Pearson Ed. France, 481p.

Martraire, C., Thiéfaine, A, Bartaguiz, D., Hiegel, F. & Fakih, H (2022) *Software craft : TDD, Clean Code et autres pratiques essentielles*, Ed : Dunod, 289 p., ISBN : 978-2-10-082520-2

Printz, J. (2010), Méthodes Agiles, Techniques de l'Ingénieur, ref. H3202.

Thomas, G., Martin, D. & Pleasants, K. (2011). Using self- and peer-assessment to enhance students' future-learning in higher education. *Journal of University Teaching & Learning Practice*, 8(1), 1-17.

Vuorikari, R., Kluzer, S. and Punie, Y., DigComp 2.2: The Digital Competence Framework for Citizens - With new examples of knowledge, skills and attitudes, EUR 31006 EN, Publications Office of the European Union, Luxembourg, 2022, ISBN 978-92-76-48883-5, doi:10.2760/490274, JRC128415.

Bibliographie du résumé long.

Code.org (2022), Maze, <https://studio.code.org/hoc/1>.

European Union-EU (2019). Key competences for lifelong learning, Eds: Luxembourg: Publications Office of the European Union, doi:10.2766/569540

Martin, C. (2009). Coder proprement, Pearson Ed. France, 481p.

Grzega, J. (2005). Learning By Teaching. The Didactic Model LdL in University Classes. Retrieved 2015, December from <http://www.joachim-grzega.de/ldl-engl.pdf>

Martraire, C., Thiéfaine, A, Bartaguiz, D., Hiegel, F. & Fakih, H (2022) Software craft : TDD, Clean Code et autres pratiques essentielles, Ed : Dunod, 289 p., ISBN : 978-2-10-082520-2

Nuninger, W. (2017). Integrated Learning Environment for blended oriented course: 3-year feedback on a skill-oriented hybrid strategy. HCI International, 9-14 July. In: Zaphiris P., Ioannou A. (eds) Learning and Collaboration Technologies. Novel Learning Ecosystems. LCT 2017. Lecture Notes in Computer Science, vol 10295. Springer, Cham. doi: 10.1007/978-3-319-58509-3_1

Nuninger, W., Hoogstoel, F. & Lancieri, L. (accepted, release 2022). A more progressive peer review to enhance formative learning: an issue of trust and motivation for commitment. In J. Sánchez-Santamaría (Eds.), *Handbook of Research on Formative and Shared Assessment to Promote Global University Learning*. Hershey, PA : IGI Global.

Printz, J. (2010), Méthodes Agiles, Techniques de l'Ingénieur, ref. H3202.

Radenkovic, D. (2012), Conduire un projet avec la méthode Scrum, Fiche pratique 0777, Techniques de l'ingénieur.

Roy, M. & Michaud, N. (2018). L'autoévaluation et l'évaluation par les pairs en enseignement supérieur : promesses et défis. *Formation et profession*, 26(2), 54-65. <http://dx.doi.org/10.18162/fp.2018.458>

Thomas, G., Martin, D. & Pleasants, K. (2011). Using self- and peer-assessment to enhance students' future-learning in higher education. *Journal of University Teaching & Learning Practice*, 8(1), 1-17.

Vuorikari, R., Kluzer, S. and Punie, Y., DigComp 2.2: The Digital Competence Framework for Citizens - With new examples of knowledge, skills and attitudes, EUR 31006 EN, Publications Office of the European Union, Luxembourg, 2022, ISBN 978-92-76-48883-5, doi:10.2760/490274, JRC128415.