



Comment concevoir et mettre en œuvre une évaluation par les pairs ?

Tanguy Pérennou (ISAE-SUPAERO)

Janvier 2025

Atelier conçu avec Nathalie Veuillez, Louis Adam (IMT Mines Albi),
Alain Bérard (INSA Toulouse) et Anne-Lise Luga (ISAE-SUPAERO)

SiUP

Université
Fédérale
Toulouse
Midi-Pyrénées

ISAE
SUPAERO

CC BY-NC-SA

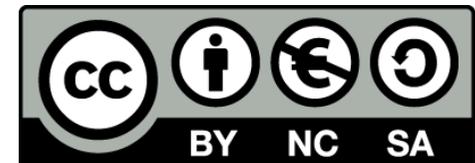
Licenses

- Creative Commons

Vous êtes libres de partager et d'adapter ce travail selon les conditions suivantes :

- **Attribution** : vous devez créditer ce travail, intégrer un lien vers la licence, et indiquer si des modifications ont été effectuées
- **Pas d'utilisation commerciale**
- **Partage dans les mêmes conditions** : si vous adaptez ce travail, vous devez diffuser le travail modifié en utilisant la même licence CC BY-NC-SA
- **Pas de restriction supplémentaire**

Voir <https://creativecommons.org/licenses/by-nc-sa/4.0>
https://wiki.creativecommons.org/wiki/Best_practices_for_attribution



- Icônes par [Font Awesome](#)

Tour de table



Présentez-vous en 1 phrase : prénom, nom, fonction, établissement, etc.

Quelles sont vos attentes par rapport à cet atelier ?



- A la fin de l'atelier, vous serez en mesure de
 - Expliquer l'apport de l'évaluation par les pairs (**EPP**) en termes d'apprentissage
 - Citer les principales étapes d'une EPP
 - Citer les principaux écueils d'une EPP
 - Planifier une première EPP avec peu d'impact

- Pour aller plus loin
 - Emparez-vous de la bibliographie
 - Faites-vous accompagner par les conseillers ou ingénieurs pédagogiques de vos établissements

1. Pourquoi une évaluation par les pairs ?
2. Processus et définitions
3. Grilles critériées
4. Rétroaction descriptive
5. Planifier une EPP



Pourquoi une évaluation par les pairs ?

Discussion : pourquoi une EPP ?



1. Individuellement (3 min)

Quel intérêt pour les étudiants ?

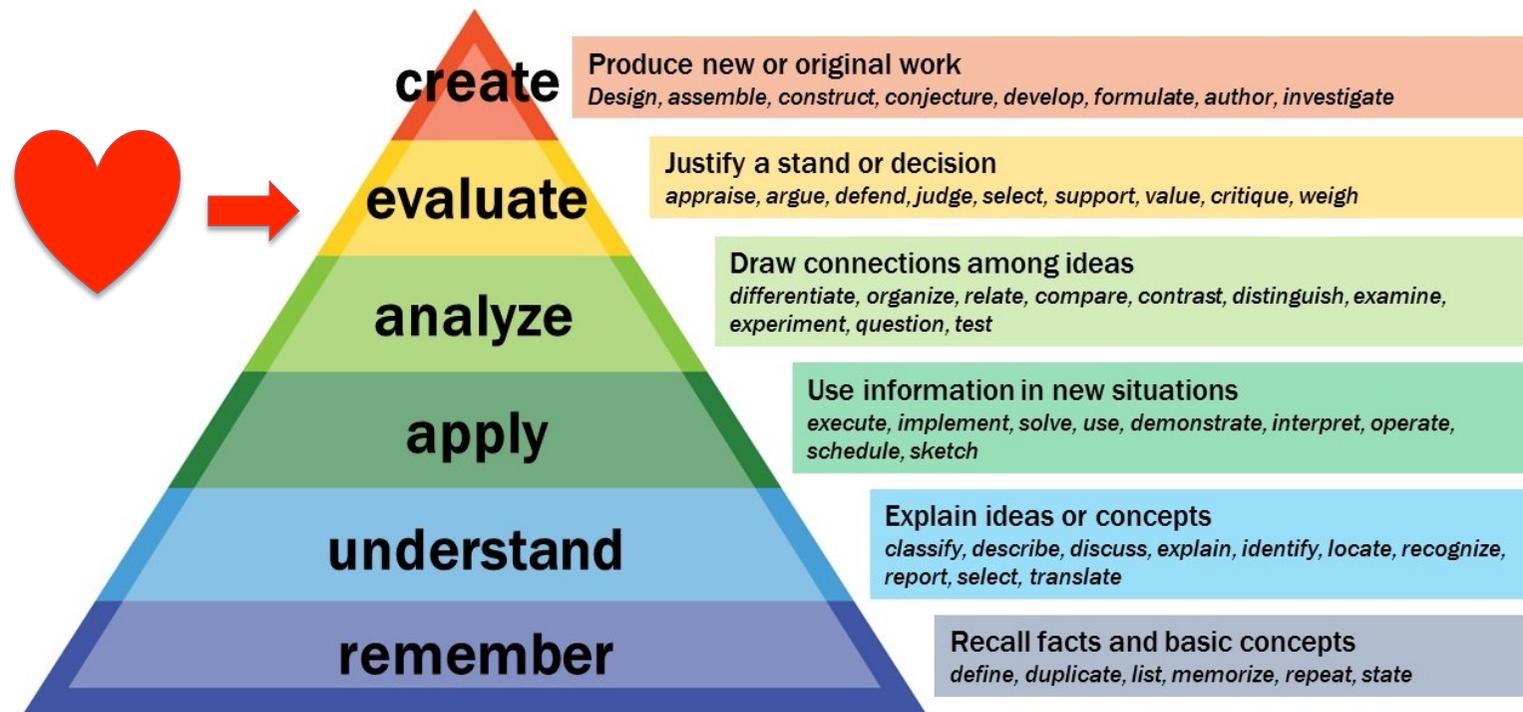
Quel intérêt pour vous ?

2. Tour de table (12 min)

Évaluer : un objectif d'apprentissage de haut niveau



Bloom's Taxonomy



Vanderbilt University Center for Teaching



Processus et définitions

Dans la peau de l'étudiant : vivez une EPP !



- Activité « Atelier Moodle »

<https://foad.univ-toulouse.fr/mod/workshop/view.php?id=10871>

1. Déposez 1 travail
2. Évaluez 3 travaux
3. Consultez les résultats

- Discussion



Exemple de processus d'évaluation par les pairs



1. Distribution d'un sujet
2. Production des travaux initiaux
3. Répartition des évaluations parmi les étudiants
4. Évaluations des travaux par les pairs
5. Évaluation des évaluations
Les évaluations ont-elles été bien faites ?
6. Calcul des notes individuelles

Ingrédients indispensables pour toute EPP



- Chaque étudiant produit un travail initial
- Chaque étudiant évalue les travaux de plusieurs autres étudiants
- Le travail de chaque étudiant est évalué par plusieurs autres étudiants

Quelles variations autour de l'EPP ?



1. Individuellement (3 min)

Quelles variations imaginez-vous dans le processus EPP ?

1. Distribution d'un sujet
2. Production des travaux initiaux
3. Répartition des évaluations parmi les étudiants
4. Évaluations des travaux par les pairs
5. Évaluation des évaluations
6. Calcul des notes individuelles

2. Tour de table (12 min)

Évaluation sommative ou formative ?



- Évaluation sommative
 - Adaptée aux MOOC
 - Barème très important mais délicat à établir
 - Attention aux collusions et conflits
- Évaluation formative (sans note ou avec note indicative)
 - Plus adaptée pour renforcer l'apprentissage
 - Moins de difficultés de barème / collusion
 - Attention aux conflits

Autres variations possibles



- Avec ou sans « Atelier Moodle »
- Travail initial : individuel ou en groupe
- Évaluation supplémentaire par l'enseignant
- Évaluation par les étudiants :
 - Compte ou ne compte pas pour valider
 - Avec ou sans évaluation des évaluations
 - Faite individuellement ou en groupe
- Préparation en amont des étudiants (ressources, activités)

Paramétrage d'un « Atelier Moodle »



- Modifier le calcul de la note finale :
 - Moyenne des évaluations du travail initial : 80% → 50%
 - Évaluation des évaluations : 20% → 50%
- Donner 5 travaux à évaluer à chacun pour avoir une moyenne significative
- Calibrer le travail pour que chaque évaluation soit réalisable en 5 à 10 minutes
- Entraînement sur des « exemples de travaux » déjà corrigés par l'enseignant



- Les étudiants sont-ils aptes à évaluer ?
 - Sauront-ils évaluer ?
 - Joueront-ils le jeu : pas de collusion, pas de superficialité ?
- Les étudiants acceptent-ils les évaluations des autres ?
 - Font-ils confiance ?
 - La formulation est-elle acceptable ?
- Peuvent-ils se passer de l'évaluation « officielle » d'un enseignant ?



- Pour pallier les problèmes de barème \Rightarrow **grille critériée**
- Pour obtenir de bons commentaires \Rightarrow **rétroaction descriptive**



Grille critériées

Outils pour évaluer



Grille critériée ou *Rubric*



Critères à évaluer
(lignes)

Niveaux de performance
(colonnes)

	★	★★	★★★	★★★★
Name		ROBER	ROBERT	Robert
Picture				
Coloring				
Writing		of Joe	ICANPIA	I can play.

Indicateurs de performance
(cases)

Par Samantha Francis, <http://owlbeteachingyou.blogspot.com>

Évaluer un travail avec une grille critériée



Pour un travail donné, dans chaque ligne, l'évaluateur choisit la case la plus appropriée

	★	★★	★★★	★★★★
Name		ROBER	ROBER	Robert
Picture				
Coloring				
Writing		o J D e	CanPIA	I can play.

Par Samantha Francis, <http://owlbeteachingyou.blogspot.com>

Dessin fait par Anaë



Exemple de grille critériée pour l'Atelier Moodle



Citez les noms des niveaux constituant la pyramide de Bloom, et indiquez à quel niveau se situe l'évaluation par les pairs.

Critères	Niveaux de performance			
	Insuffisant 0	Faible 1	Visé 2	Au-delà 3
Niveaux	Aucun niveau correct n'est cité	Niveaux cités partiellement ou dans le désordre	Les 6 niveaux sont cités dans l'ordre	Visé + quelques exemples
Situation de l'EPP	EPP non située dans les niveaux	EPP située dans les niveaux inférieurs	EPP située dans les niveaux supérieurs	EPP située précisément au niveau Évaluation

Quelques recommandations



- 6 critères ou moins
 - Mesurables
 - Indépendants
 - Déduits des objectifs d'apprentissage
- Limitez-vous à 3 ou 4 niveaux
 - Un niveau « visé »
 - Pas de niveau « moyen »
 - Le niveau « maximum » doit être atteignable
- Doit tenir sur une feuille A4 en étant lisible

Quelques recommandations



- Rédigez les indicateurs en partant de “visé” (ou “au-delà” si vous débutez)
- Testez votre grille
- Faites-la relire

Comment obtenir une note avec cette grille ?



Critère	Poids	Niveaux de performance			
		Insuffisant 0	Faible 1	Visé 3	Au-delà 4
Maîtriser l'utilisation de Code::Blocks et l'environnement de travail	1	Les étudiants ne peuvent pas maîtriser l'environnement	Les étudiants ont besoin d'aide fréquente pour maîtriser l'environnement	Les étudiants savent utiliser l'outil de développement Code::Blocks et organiser leur espace de travail	Les étudiants aident les autres binômes pour l'usage de leurs environnements
Comprendre le langage vu depuis le début du cours	2	Les notions antérieures ne sont pas maîtrisées	Les notions de la séance ne sont pas maîtrisées	Les étudiants sont en phase avec les notions vues	Les étudiants sont en avance sur les séances
Corriger les erreurs avec les messages du compilateur	2	Les étudiants ne savent pas localiser leurs erreurs	Les étudiants ne comprennent pas leurs erreurs : corrigent par tâtonnement ou avec de l'aide extérieure ou avec des warnings encore présents	Les étudiants comprennent leurs erreurs et aboutissent à un code sans erreur et sans warning	Les étudiants sont autonomes pour comprendre et corriger leurs erreurs rapidement
Réaliser un programme conforme au cahier des charges de l'exercice	2	Le programme plante ou ne fait pas ce qui est demandé ET les étudiants ne savent pas localiser leurs erreurs	Le programme s'exécute partiellement	Les étudiants obtiennent un programme s'exécutant conformément au cahier des charges	Les étudiants sont autonomes pour tester, gérer les cas particuliers, optimiser leur programme
Réaliser un code compréhensible et réexploitable	1	Le programme est difficilement compréhensible par un lecteur extérieur au binôme	Les noms des identificateurs ne sont pas explicites ou pas de commentaires ou l'indentation n'est pas cohérente	Le code est clair : indentation, nom des identificateurs explicites, code commenté.	Le code est complet : les entêtes ET les commentaires sont concis et pertinents
Déposer des documents en ligne sur Chamilo dans les délais	2	Pas de dépôts	Les exercices sont partiellement déposés ou mal organisés	Les exercices de la séance sont tous exploitables et déposés	Les exercices sont tous déposés ET les exercices complémentaires sont faits.

Sources :

Stéphane Guillet et al., ENSE3 1A, Algorithmique et Programmation



Poids des critères, valeurs des niveaux



Critère	Poids	Niveaux de performance			
		Insuffisant 0	Faible 1	Visé 3	Au-delà 4
Maîtriser l'utilisation de Code::Blocks et l'environnement de travail	1	Les étudiants ne peuvent pas maîtriser l'environnement	Les étudiants ont besoin d'aide fréquente pour maîtriser l'environnement	Les étudiants savent utiliser l'outil de développement Code::Blocks et organiser leur espace de travail	Les étudiants aident les autres binômes pour l'usage de leurs environnements
Comprendre le langage vu depuis le début du cours	2	Les notions antérieures ne sont pas maîtrisées	Les notions de la séance ne sont pas maîtrisées	Les étudiants sont en phase avec les notions vues	Les étudiants sont en avance sur les séances
Corriger les erreurs avec les messages du compilateur	2	Les étudiants ne savent pas localiser leurs erreurs	Les étudiants ne comprennent pas leurs erreurs : corrigent par tâtonnement ou avec de l'aide extérieure ou avec des warnings encore présents	Les étudiants comprennent leurs erreurs et aboutissent à un code sans erreur et sans warning	Les étudiants sont autonomes pour comprendre et corriger leurs erreurs rapidement
Réaliser un programme conforme au cahier des charges de l'exercice	2	Le programme plante ou ne fait pas ce qui est demandé ET les étudiants ne savent pas localiser leurs erreurs	Le programme s'exécute partiellement	Les étudiants obtiennent un programme s'exécutant conformément au cahier des charges	Les étudiants sont autonomes pour tester, gérer les cas particuliers, optimiser leur programme
Réaliser un code compréhensible et réexploitable	1	Le programme est difficilement compréhensible par un lecteur extérieur au binôme	Les noms des identificateurs ne sont pas explicites ou pas de commentaires ou l'indentation n'est pas cohérente	Le code est clair : indentation, nom des identificateurs explicites, code commenté.	Le code est complet : les entêtes ET les commentaires sont concis et pertinents
Déposer des documents en ligne sur Chamilo dans les délais	2	Pas de dépôts	Les exercices sont partiellement déposés ou mal organisés	Les exercices de la séance sont tous exploitables et déposés	Les exercices sont tous déposés ET les exercices complémentaires sont faits.

Sources :
Stéphane Guillet et al., ENSE3 1A, Algorithmique et Programmation



Note max = somme des poids × valeur max



Critère	Poids	Niveaux de performance			
		Insuffisant 0	Faible 1	Visé 3	Au-delà 4
Maîtriser l'utilisation de Code::Blocks et l'environnement de travail	1	Les étudiants ne peuvent pas compiler leur code	Les étudiants ont besoin de l'aide de l'enseignant pour compiler leur code	Les étudiants savent utiliser Code::Blocks et organiser leur espace de travail	Les étudiants aident les autres à compiler leur code
Comprendre le langage vu depuis le début du cours	2	Les notions antérieures ne sont pas maîtrisées	Les notions de la séance ne sont pas maîtrisées	Les étudiants sont en phase avec les notions vues	Les étudiants sont en avance sur les séances
Corriger les erreurs avec les messages du compilateur	2	Les étudiants ne savent pas localiser leurs erreurs	Les étudiants ne comprennent pas leurs erreurs : corrigent par tâtonnement ou avec de l'aide extérieure ou avec des warnings encore présents	Les étudiants comprennent leurs erreurs et aboutissent à un code sans erreur et sans warning	Les étudiants sont autonomes pour comprendre et corriger leurs erreurs rapidement
Réaliser un programme conforme au cahier des charges de l'exercice	2	Le programme plante ou ne fait pas ce qui est demandé ET les étudiants ne savent pas localiser leurs erreurs	Le programme s'exécute partiellement	Les étudiants obtiennent un programme s'exécutant conformément au cahier des charges	Les étudiants sont autonomes pour tester, gérer les cas particuliers, optimiser leur programme
Réaliser un code compréhensible et réexploitable	1	Le programme est difficilement compréhensible par un lecteur extérieur au binôme	Les noms des identificateurs ne sont pas explicites ou pas de commentaires ou l'indentation n'est pas cohérente	Le code est clair : indentation, nom des identificateurs explicites, code commenté.	Le code est complet : les entêtes ET les commentaires sont concis et pertinents
Déposer des documents en ligne sur Chamilo dans les délais	2	Pas de dépôts	Les exercices sont partiellement déposés ou mal organisés	Les exercices de la séance sont tous exploitables et déposés	Les exercices sont tous déposés ET les exercices complémentaires sont faits.

Somme poids = 10

Valeur max = 4

Note max = 40

Sources :

Stéphane Guillet et al., ENSE3 1A, Algorithmique et Programmation



Note d'un travail = somme pondérée cases cochées



Critère	Poids	Niveaux de performance			
		Insuffisant 0	Faible 1	Visé 3	Au-delà 4
Maîtriser l'utilisation de Code::Blocks et l'environnement de travail	1	Les étudiants ne peuvent pas maîtriser l'environnement	Les étudiants ont besoin d'aide fréquente pour maîtriser l'environnement	Les étudiants savent utiliser l'outil de développement Code::Blocks et organisent leur espace de travail ✓	Les étudiants aident les autres binômes pour l'usage de leurs environnements
Comprendre le langage vu depuis le début du cours	2	Les notions antérieures ne sont pas maîtrisées	Les notions de la séance ne sont pas maîtrisées ✓	Les étudiants sont en phase avec les notions vues	Les étudiants sont en avance sur les séances
Corriger les erreurs avec les messages du compilateur	2	Les étudiants ne savent pas localiser leurs erreurs	Les étudiants ne comprennent pas leurs erreurs : corrigent par tâtonnement ou avec de l'aide extérieure ou avec des warnings encore présents	Les étudiants comprennent leurs erreurs et aboutissent à un code sans erreur et sans warning ✓	Les étudiants sont autonomes pour comprendre et corriger leurs erreurs rapidement
Réaliser un programme conforme au cahier des charges de l'exercice	2	Le programme plante ou ne fait pas ce qui est demandé ET les étudiants ne savent pas localiser leurs erreurs	Le programme s'exécute partiellement	Les étudiants obtiennent un programme s'exécutant conformément au cahier des charges	Les étudiants sont autonomes pour tester, gérer les cas particuliers, optimiser leur programme ✓
Réaliser un code compréhensible et réexploitable	1	Le programme est difficilement compréhensible par un lecteur extérieur au binôme ✓	Les noms des identificateurs ne sont pas explicites ou pas de commentaires ou l'indentation n'est pas cohérente	Le code est clair : indentation, nom des identificateurs explicites, code commenté.	Le code est complet : les entêtes ET les commentaires sont concis et pertinents
Déposer des documents en ligne sur Chamilo dans les délais	2	Pas de dépôts ✓	Les exercices sont partiellement déposés ou mal organisés	Les exercices de la séance sont tous exploitables et déposés	Les exercices sont tous déposés ET les exercices complémentaires sont faits.

19 sur 40

$$1 \times 3$$

$$+ 2 \times 1$$

$$+ 2 \times 3$$

$$+ 2 \times 4$$

$$+ 1 \times 0$$

$$+ 2 \times 0$$

Obtenez les notes sur 40 pour ces 2 travaux



Critère	Poids	Niveaux de performance			
		Insuffisant 0	Faible 1	Visé 3	Au-delà 4
Maîtriser l'utilisation de Code::Blocks et l'environnement de travail	1	Les étudiants ne peuvent pas maîtriser l'environnement	Les étudiants ont besoin d'aide fréquente pour maîtriser l'environnement	Les étudiants savent utiliser l'outil de développement Code::Blocks et organiser leur espace de travail ✓	Les étudiants aident les autres binômes pour l'usage de leurs environnements
Comprendre le langage vu depuis le début du cours	2	Les notions antérieures ne sont pas maîtrisées	Les notions de la séance ne sont pas maîtrisées	Les étudiants sont en phase avec les notions vues ✓	Les étudiants sont en avance sur les séances
Corriger les erreurs avec les messages du compilateur	2	Les étudiants ne savent pas localiser leurs erreurs	Les étudiants ne comprennent pas leurs erreurs : corrigent par tâtonnement ou avec de l'aide extérieure ou avec des warnings encore présents	Les étudiants comprennent leurs erreurs et aboutissent à un code sans erreur et sans warning ✓	Les étudiants sont autonomes pour comprendre et corriger leurs erreurs rapidement
Réaliser un programme conforme au cahier des charges de l'exercice	2	Le programme plante ou ne fait pas ce qui est demandé ET les étudiants ne savent pas localiser leurs erreurs	Le programme s'exécute partiellement	Les étudiants obtiennent un programme s'exécutant conformément au cahier des charges ✓	Les étudiants sont autonomes pour tester, gérer les cas particuliers, optimiser leur programme
Réaliser un code compréhensible et réexploitable	1	Le programme est difficilement compréhensible par un lecteur extérieur au binôme	Les noms des identificateurs ne sont pas explicites ou pas de commentaires ou l'indentation n'est pas cohérente	Le code est clair : indentation, nom des identificateurs explicites, code commenté.	Le code est complet : les entêtes ET les commentaires sont concis et pertinents ✓
Déposer des documents en ligne sur Chamilo dans les délais	2	Pas de dépôts	Les exercices sont partiellement déposés ✓	Les exercices de la séance sont tous exploitables et déposés	Les exercices sont tous déposés ET les exercices complémentaires sont faits.

Sources :
Stéphane Guillet et al., ENSE3 1A, Algorithmique et Programmation

Critère	Poids	Niveaux de performance			
		Insuffisant 0	Faible 1	Visé 3	Au-delà 4
Maîtriser l'utilisation de Code::Blocks et l'environnement de travail	1	Les étudiants ne peuvent pas maîtriser l'environnement	Les étudiants ont besoin d'aide fréquente pour maîtriser l'environnement ✓	Les étudiants savent utiliser l'outil de développement Code::Blocks et organiser leur espace de travail	Les étudiants aident les autres binômes pour l'usage de leurs environnements
Comprendre le langage vu depuis le début du cours	2	Les notions antérieures ne sont pas maîtrisées	Les notions de la séance ne sont pas maîtrisées ✓	Les étudiants sont en phase avec les notions vues	Les étudiants sont en avance sur les séances
Corriger les erreurs avec les messages du compilateur	2	Les étudiants ne savent pas localiser leurs erreurs	Les étudiants ne comprennent pas leurs erreurs : corrigent par tâtonnement ou avec de l'aide extérieure ou avec des warnings encore présents ✓	Les étudiants comprennent leurs erreurs et aboutissent à un code sans erreur et sans warning	Les étudiants sont autonomes pour comprendre et corriger leurs erreurs rapidement
Réaliser un programme conforme au cahier des charges de l'exercice	2	Le programme plante ou ne fait pas ce qui est demandé ET les étudiants ne savent pas localiser leurs erreurs	Le programme s'exécute partiellement	Les étudiants obtiennent un programme s'exécutant conformément au cahier des charges ✓	Les étudiants sont autonomes pour tester, gérer les cas particuliers, optimiser leur programme
Réaliser un code compréhensible et réexploitable	1	Le programme est difficilement compréhensible par un lecteur extérieur au binôme ✓	Les noms des identificateurs ne sont pas explicites ou pas de commentaires ou l'indentation n'est pas cohérente	Le code est clair : indentation, nom des identificateurs explicites, code commenté.	Le code est complet : les entêtes ET les commentaires sont concis et pertinents
Déposer des documents en ligne sur Chamilo dans les délais	2	Pas de dépôts ✓	Les exercices sont partiellement déposés ou mal organisés	Les exercices de la séance sont tous exploitables et déposés	Les exercices sont tous déposés ET les exercices complémentaires sont faits.

Sources :
Stéphane Guillet et al., ENSE3 1A, Algorithmique et Programmation

Grille analytique vs single-point



Analytique :
les indicateurs sont **tous** écrits

	★	★★	★★★	★★★★
Name		ROBER	ROBERT	Robert
Picture				
Coloring				
Writing		o J e	ICANPIA	I can play.

By Samantha Francis, <http://owlbeteachingyou.blogspot.com>

Single-point :
indicateurs **visés** seulement

	★	★★	★★★	★★★★
Name			ROBERT	
Picture				
Coloring				
Writing			ICANPIA	

Évaluer avec une grille single-point



- Si le niveau « Visé » est atteint : l'entourer
- Autres niveaux : écrire un commentaire spécifique pour chaque travail (possiblement plusieurs lignes)



	★	★★	★★★	★★★★
Name			ROBERT	
Picture				Détails : papillons, ...
Coloring				Bien rempli
Writing			ICANPIA	

Par Samantha Francis, <http://owlbeteachingyou.blogspot.com>





Rétroaction descriptive

Outils pour évaluer

SiUP

Université
Fédérale
Toulouse
Midi-Pyrénées

ISAB
SUPAERO

CC BY-NC-SA

Comment communiquer une note ?



- Tour de table :
 - Comment communiquez-vous une note à un étudiant ?

Dans une EPP, les notes ne suffisent pas



- Quelle validité accorder aux notes reçues des pairs ?
- Doivent être complétées par une **rétroaction descriptive**
 - Ce qui a été accompli, en quoi une réponse est appropriée
 - Pistes d'amélioration
- Bonus : effet positif sur l'apprentissage

Effets des signaux envoyés par une rétroaction

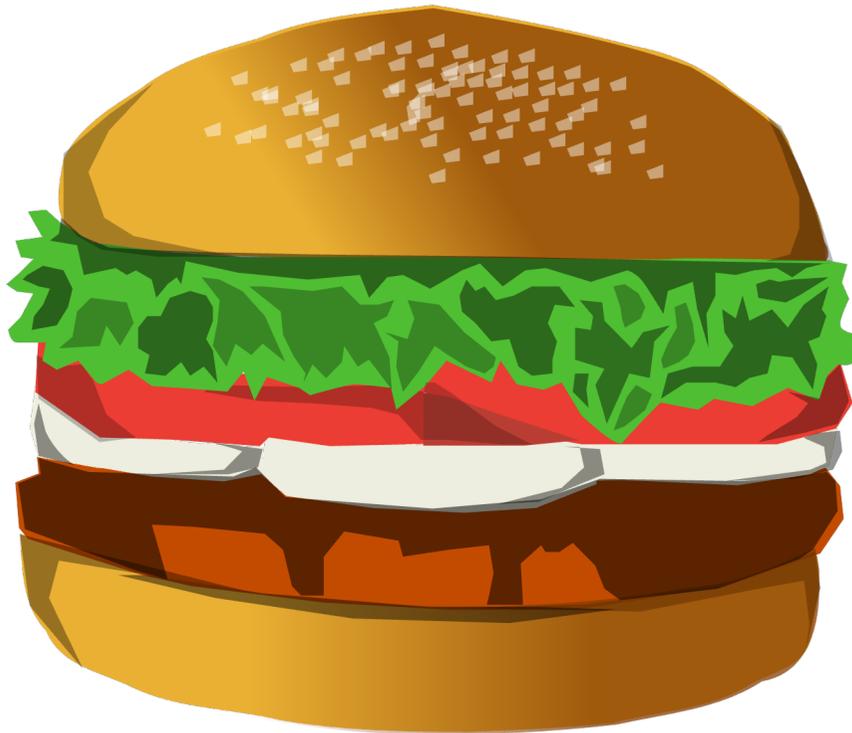


À inclure



Signaux	Progrès possibles	Confiance
Positif sur la personne	+ +	+ + +
Positif sur la production	+ + +	+
Négatif sur la production	+ +	-
Négatif sur la personne	- -	- - -
<i>(pas de rétroaction)</i>	-	-

Allégorie de rétroaction descriptive



Première tranche de pain :
éléments objectifs et positifs

Steak :
**pistes d'amélioration,
remarques sur les erreurs**

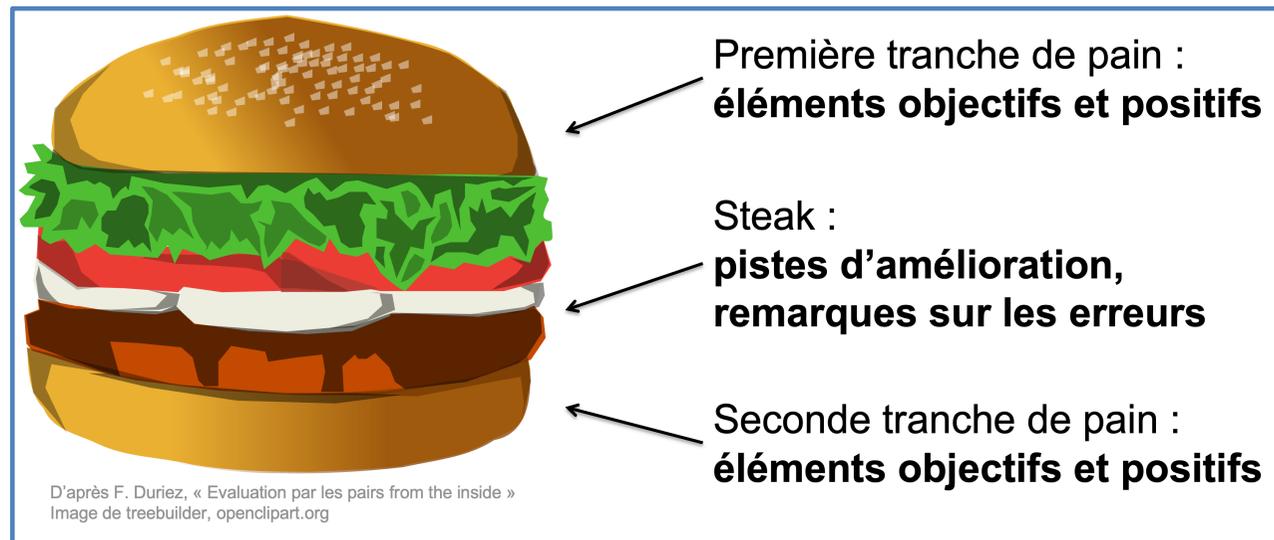
Seconde tranche de pain :
éléments objectifs et positifs

D'après F. Duriez, « Évaluation par les pairs from the inside »
Image de treebuilder, openclipart.org

Formulez une rétroaction descriptive



- En binôme, *formulez par écrit une rétroaction descriptive de 3 lignes sur la qualité et l'efficacité de la diapo précédente :*

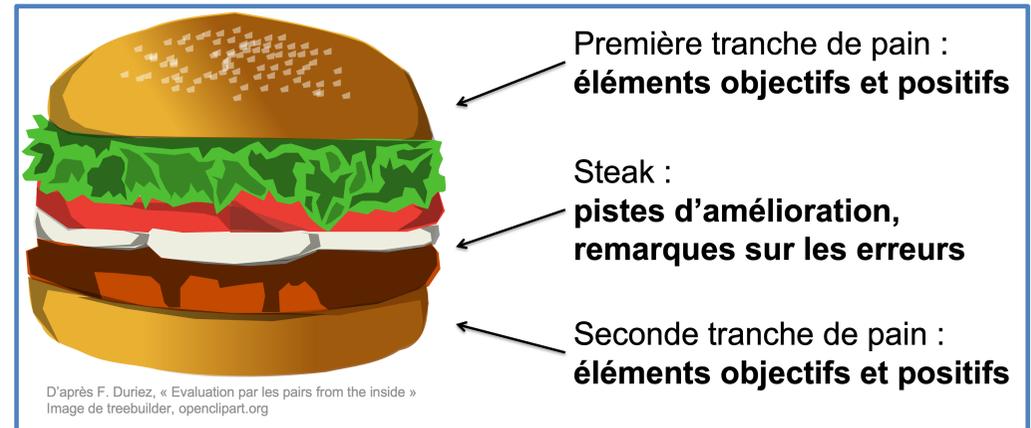


- Tour de table : partagez vos rétroactions

Exemple de RD sur la qualité d'une diapo



- Cette diapositive contient une illustration bien créditée, avec une légende et une animation convaincantes
- Elle pourrait mieux faire le lien avec les signaux envoyés par une rétroaction (diapo 35), voir par exemple la référence Bachelet 2014.
- L'illustration est bien choisie pour le public visé. C'est un de tes points forts, continue comme ça !





Planifier une évaluation par les pairs

Prévoir du temps



- Pour préparer soigneusement votre grille critériée
 - Quels sont vos objectifs d'apprentissage ?
 - Comment les mesurer ?
 - Prévoir des relecteurs
- Pour entraîner vos élèves à
 - Évaluer
 - Utiliser une grille critériée
 - Formuler des rétroactions descriptives
- Pour évaluer quelques travaux si nécessaire
- Pour créer un « Atelier Moodle » si souhaité

Vous hésitez ?



- Commencez par une évaluation **formative**
- ... à **petite échelle**
 - Peu d'élèves
 - Travaux de petite taille
 - Partie non essentielle/critique
- ... **encadrée**
 - Commencez par des « travaux typiques » anciens
 - En face à face
 - Fournissez un format-type pour tendre vers des rétroactions descriptives

Références évaluation par les pairs



- Amaury Daele, *Évaluer ou être évalué, telle est la question*, blog (2009)
<https://pedagogieuniversitaire.wordpress.com/2009/08/30/evaluer-ou-etre-evalue-telle-est-la-question>
- Nathalie Veuillez, *Minute de l'innovation pédagogique : l'évaluation entre pairs à IMT Mines Albi* (2017)
<https://www.innovation-pedagogique.fr/article2874.html>
- Rémi Bachelet, *L'évaluation par les pairs en projet* (2011)
- Rémi Bachelet, *Évaluation par les pairs sandwich*, slideshare (2014)
<https://pt.slideshare.net/bachelet/evaluation-par-les-pairs-sandwich-29865880>
- Frédéric Duriez, *Évaluation par les pairs from the inside*, blog (2013)
<https://didac2b.wordpress.com/2013/10/27/evaluation-par-les-pairs-from-the-inside>
- Nicol & MacFarlane-Dick, *Formative assessment and self-regulated learning: A model and seven principles of good feedback practice* (2005)

Références grilles critériées



- Amaury Daele, *Les grilles d'évaluation critériées*, blog (2010), <https://pedagogieuniversitaire.wordpress.com/2010/03/22/les-grilles-devaluation-criteriees/>
- Jennifer Gonzalez, *Know Your Terms: Holistic, Analytic, and Single-Point Rubrics*, <https://www.cultofpedagogy.com/holistic-analytic-single-point-rubrics> (2014)
- Julien Douady et al., *Une grille critériée générique pour l'évaluation des compétences expérimentales des étudiants à l'université* (2012)
- Kathleen Montgomery, *Classroom Rubrics: Systematizing What Teachers Do Naturally* (2000)
- FA2L, *L'évaluation des apprentissages*, <https://www.fa2l.be/les-cahiers-de-fa2l> (2015)

Ressources en ligne sur Moodle

<https://foad.univ-toulouse.fr/course/view.php?id=666>





Merci