Teamwork assessment and peerwise scoring: Combining process and product assessment

Ian G. Kennedy[1], Paul H. Vossen[2]

Abstract: Teamwork is commonly required in the industry [Sa95]. At university, students learn teamwork by working in groups, for example during a software development assignment. Teamwork affects their productivity. Lecturers evaluate their overall group performance. Lecturers also need to assess the cooperation of each team member. Cooperation data comes from the students involved, e.g., via a questionnaire. Here we draw attention to specific items for assessment. We then show how to merge or aggregate evaluations using the lecturer's overall quality scale and the students' peer assessment scale. To this end, we will explain and demonstrate the underlying Split-Join Invariance principle using two compatible scoring formulae.

Keywords: Grading, Peer Assessment, Performance Rating, Quality Scale, Scoring Rule, Scoring Rubric, Split-Join Invariance (SJI), Team Cooperation

1 Design of assessment for teamwork

Think of this realistic scenario. A lecturer has a software engineering class of 35 students. They have to work in 8 groups of 3 or 4 students. There is no final exam, e.g. a multiple-choice test, just the outcome of teamwork. Scores are to be reported on an individual basis, though, and the amount of assessment will be about a half hour per team. How to assess doing justice to all involved? Students who are grade grubbing manipulate overall impression marks easily, and there may be free riders as well. A fair and robust assessment will not be challenged by students. A fair assessment will use attributes which are made known beforehand, along with the rule for combining the attributes, and any weights thereof. The specified rules must be consistently applied to all students and groups, and be traceable. Moreover, arbitrariness, discrimination or dishonesty should be avoided by awarding the marks mechanistically, according to publicised norms. Finally, the marks should not pretend to be more accurate than they are; their production can be replicated.

In designing courses, e.g., practice-oriented software engineering courses, we consider challenges involved in preparing students for the real world. After graduation, students will be required to work in teams, e.g., to build large software systems [Sa95]. Thus, they

[1] Ex University of the Witwatersrand, Johannesburg, South Africa, 2000, dr.iankennedy@gmail.com
[2] Research Institute SQUIRE, Kiefernweg 1A, Niederstetten, D-97996, Germany, p.h.vossen@googlemail.com

must now learn to work in teams. A textbook knowledge of teamwork assessed by a written exam on it is clearly insufficient*: students learn teamwork only by doing it*. Students need to reflect and evaluate their team cooperation in a self-critical way. Lecturers –in addition to their assessment of the overall team deliverable– need detailed and reliable information about the teamwork. Such data can only come from the students, e.g., through questionnaires or by automated measurements during or after group work. Our questions are: What data is required for appropriate, meaningful peer and team assessment? How can the results of the peer assessment be combined with the assessment of the productivity and products of the group work into a final evaluation of each individual student?

2 Attributes in the PROCESS scoring rubric

After finding only inferior references, we invented our own list of attributes of cooperation (Table 1, based on [VK17]) that we feel all team members should learn. These attributes can be revised, localised and then taught during introductory lectures. They can be tested and assessed via a group project. Here are our proposed attributes (scoring rubrics):

Outcome no.	PROCESS Learning Objective
1	Commitment
2	Collaboration
3	Coaching each other
4	Coordination
5	Control
6	Communication
7	Climate fostering
8	Constructive criticism
9	Coping with adversity and diversity
10	Conflict management

Tab. 1: Specimen individual PROCESS rubric

Commitment: It is easy to overlook the importance of commitment. Team members must see the project goal or the corporate goal and remember that they pass, or are only paid a salary to produce software successfully and align their allegiance to this aim.
Collaboration: Collaboration is working jointly towards the agreed-upon project goal. How well does each team member facilitate the work of others in the team?
Coaching each other: Team members must learn from each other and teach each other.
Coordination: Software team members have to coordinate their activities to ensure, e.g., that they do not save their work on top of others' work.
Control: An airline pilot takes the final responsibility for the control of the aeroplane. Although the co-pilot can fly the plane, only the pilot has a hand on the joystick in the cockpit. So too, a chain of command in the software team will develop.
Communication: Communication includes listening, and written as well as spoken com-

munication. It includes upward, downward and horizontal communication inside and outside the group.
Climate fostering: Team members should help each other to maintain an open and positive team climate for discussions.
Constructive criticism: It is important to help others with constructive criticism.
Coping with adversity and diversity: Coping with difficult circumstances includes, e.g., coping with absenteeism and change requests.
Conflict management: In practice, there always will be conflict in a team. The question is: how well does the team member overcome the conflicts and resolve the issues?

3 Attributes in the PRODUCT scoring rubrics

Table 2 shows our specimen rubric for software development [VK17]. We do not propose a final or standard list of assessment criteria. We include the specimen here to make our paper more readable and practice-oriented. The list of PRODUCT outcomes in Table 2 is just an example because it is highly dependent on the application area as to which attributes will be assessed. The PRODUCT is a software application including documentation. Each item in the table has a deliverable attached to it, which the lecturer will evaluate.

Outcome no.	PRODUCT Learning Objective
1	Purpose of project
2	Project specification
3	Library review
4	Functionality
5	Usability
6	Portability
7	Maintainability
8	Documentation
9	Timeliness
10	References

Tab. 2: Specimen group PRODUCT rubric

4 Technical aspects

If the lecturer uses peer assessment only for formative evaluation, then there is no need to go beyond simple questionnaire-based scoring rubrics. In practice, however, peer assessment is frequently used to produce individual student scores, i.e., for summative evaluation. The peer assessment procedure outlined above can inform the lecturer about team dynamics. Such data evaluate the individual student's cooperation. Appropriate rating methods, e.g. Likert scales, including possible weighting of assessment criteria or rubrics can be used to turn the captured data into quantitative statements. In the next section, we examine the necessary and sufficient conditions for such scoring formulae to yield a sound, robust and fair overall evaluation of each student involved in a team.

4.1 Split-Join-Invariance (SJI)

The most important principle for consistent, robust and fair peer assessment is called Split-Join-Invariance. See also [Vo11, Vo14, Vo17, VK17]. If we split the overall team score into single student scores, and then join those student scores by a suitable aggregation or averaging function, e.g. the arithmetic or geometric mean, the result must be equal to the initial team score given by the lecturer. *Older models for summative peer assessment missed or ignored this simple but powerful requirement*. There are two reasons for requir-ing SJI. The first is a conceptual reason, and can be explained by simply asking: *what else could we mean with overall team score as an aggregated or averaged evaluation of the joint work of all team members?* The point is that there is no *a priori* guarantee that the calculated average of student scores will equal the lecturer's initial team score. We have to build this into our scoring formula. The second reason is equally important. Let us ask again a simple question*: what could happen if we were not careful in our choice of a scoring formula so that the mean student score was not equal to the overall team score from which the student scores were derived?* If the mean student score is lower than the team score, then some students may rightly complain that they do not get full credit for their involvement in the joint work. In the case of a higher mean student score, the students might demand that their scores be recalculated based on this higher mean score. So, for consistency, plausibility, acceptability and accountability reasons, every peer assessment scoring formula – based on the arithmetic or geometric or any other plausible mean – must satisfy the SJI principle.

4.2 SJI-incompatible versus SJI-compatible approaches

Our [VK17] extensive literature research covering more than 25 years of publications revealed that no scoring formula proposed for peer assessment took SJI or a similar principle into account. Instead, scoring rules were proposed based on their *intuitive* plausibility or simplicity, not for reasons of logical or mathematical *appropriateness* like correctness, consistency, robustness and fairness. Previous peer-assessment scoring formulae fell into the category of linear functions in two variables: a *team score* and a *student rating.* Authors dealt mainly with the question: how can the two values (a rating and a score) be combined into a single value (score) by adding or multiplying, with or without weighting? We know of only one proposal [Ne12] which goes beyond this simple framework and adds a quadratic (or *parabolic*, in the terminology of the author) component to prevent individual scores from growing too large. All such solutions (except [Ne12]) are flawed in that final student scores can transcend the boundaries of the scoring scale. In all, the average student score need not equal the initial team score set by the lecturer, violating our Split-Join-Invariance principle [Vo17].

There are two main classes of SJI-compatible scoring rules. One class adopts the arithmetic mean (AM) and the other one the geometric mean (GM) for averaging student scores. Whereas there is only one generalised scoring scheme associated with GM, there are (at least) two distinct scoring schemes within the class of AM-based scoring rules: symmetric

versus asymmetric scoring formulae. Each class or subclass has a single scoring rule, i.e., a non-linear scoring function with one or two parameters. For space reasons, we will only sketch the two general scoring formulae for the class of arithmetic scoring functions. Let us first introduce the terminology and symbols to be used. The team score based on lecturer's assessment of the team's final PRODUCT is *t*. A student's PROCESS rating based on peer assessments by all peers in his or her team is σ_i for student *i*. It is a number between -*k* and +*k* on a bipolar Likert scale (a generalisation to the entire real line will appear in a forthcoming paper). The degree to which this student rating is to have an impact on the team score will be denoted by $\varphi(t,r)$, $\varphi_{t,r}$ or just φ. Here, *t* is the team score as introduced above; *r* is an integer or real number indicating faculty's or lecturer's tolerance or intolerance regarding the peer assessment impact. For instance, in the AM-case $r = 0$ corresponds to a medium tolerance level. Finally, there is a unique formula for each class or subclass which combines all these ingredients to produce a SJI-compatible student score s_i within the range of scores, i.e., between 0 and 1. The key ingredient is the expression for $\varphi_{t,r}$, which is different for symmetric or asymmetric AM-based scoring functions.

4.3 Symmetric scoring formula

Our [VK17] symmetric AM-based scoring formula has the following form, where τ is defined on the bipolar Likert rating scale -*k* … +*k* and denotes the overall team rating:

$$s_i = \begin{cases} t + t \times \frac{1}{1+t^r} \times \frac{\sigma_i - \tau}{k+\tau}, & \sigma_i < \tau \\ t + (1-t) \times \frac{1}{1+t^r} \times \frac{\sigma_i - \tau}{k-\tau}, & \sigma_i \geq \tau \end{cases} \quad (1)$$

Score s_i can never escape the valid range of scores [0,1]. The tolerance parameter *r* can be any positive or negative real number. Taking $r = 0$ gives a φ of 0.5, representing a mediocre tolerance level. Positive *r* means a high(er) tolerance $\varphi > 0.5$; negative r means a low(er) tolerance $\varphi < 0.5$. In practice, integer values may be taken for *r*, e.g. values between -10 and +10. Taking $r = 1$ or 2 gives easy to handle, tolerant φ's. Taking $r = -1$ or -2 is suitable for the less tolerant mindset. It is possible to transform (1) into special, simpler looking formulae if required. What about τ? It should be such that the SJI principle holds. It turns out that τ is equal to the ψ-arithmetic mean of student rates σ_i for ψ equal to the scoring function (1).

4.4 Asymmetric scoring formula

Our [VK17] general asymmetric AM-based scoring formula has the following structure, where τ is the same as above and tolerance $r \geq 1$:

$$s_i = \begin{cases} t + t \times (1 - t^{r-1}) \times \frac{\sigma_i - \tau}{k+\tau}, & \sigma_i < \tau \\ t + (1-t) \times (1 - (1-t)^{r-1}) \times \frac{\sigma_i - \tau}{k-\tau}, & \sigma_i \geq \tau \end{cases} \quad (2)$$

The most important point about the asymmetric arithmetic scoring formula is that low student ratings ($\sigma_i < \tau$) are handled differently from high student ratings ($\sigma_i > \tau$). If we increase *r*, the tolerance is increased and vice versa. For $r = 1$, we effectively prohibit peer ratings from having any influence on the student score. An exceptional case occurs for $r = 2$: it is the only case in which lecturer's team score falls within a range of tolerance which has the same length $t \times (1\text{-}t)$ left of t as right from t. We call this unique case the *balanced tolerance range scoring formula* (not to be confused with a symmetric scoring formula). Again, it turns out that τ equals the ψ-arithmetic mean of student rates σ_i for ψ equal to the scoring function (2).

5 Discussion and Conclusion

This paper has raised awareness about the categories and items that need to be assessed for teamwork results and showed how we could get student ratings for the cooperation, e.g., in a software team, based on peer ratings by other members of the team. It has also given a scoring rubric for the lecturer to mark the group. Finally, the work has shown how to consistently, robustly and fairly combine these into an aggregated evaluation for the student's reports, all based on the important Split-Join-Invariance principle. We also provided some example scoring functions.

References

[Ne12] Nepal, K. P.: An approach to assign individual marks from a team mark: the case of Australian grading system at universities. Assessment & Evaluation in Higher Education, 37/5, S. 555-562, 2012.

[Sa95] Saeki, M.: Communication, collaboration and cooperation in software development: how should we support group work in software development? In: IEEE Software Engineering Conference, Asia Pacific, S. 12-20, 1995.

[VK17] Vossen, P.; Kennedy, I.: A fair group marking scheme combining process and product assessment, Forthcoming in Practitioner Research in Higher Education Conference, University of Cumbria, Manchester, S. 14, 2017.

[Vo11] Vossen, P. H.: A truly generic performance assessment scoring system. In (Gómez Chova, L.; Candel Torres, I.; López Martínez, A., Hrsg.): Proc. of the INTED 2011 Conference, Valencia - Spain, March 7-9, S. 2448-2458, 2011.

[Vo14] Vossen, P. H.: Educational Assessment Engineering: A Pattern Approach. In (Balas, V.E.; Jain L.C.; Kovačević B., Hrsg.): Soft Computing Applications, Springer International Publishing, S. 605-619, 2014.

[Vo17] Vossen, P. H.: Distributive Fairness in Educational Assessment: Psychometric Theory meets Fuzzy Logic. In (Balas et. al., Hrsg.): Proceedings of the SOFA 2016 Conference. Springer Advances in Intelligent Systems and Computing, Springer International Publishing, S. 17, 2017 (in press).