Reliability in the Assessment of Program Quality by Teaching Assistants During Code Reviews

Michael James Scott
Department of Computer Science
Brunel University London
United Kingdom
michael.scott@brunel.ac.uk

Gheorghita Ghinea
Department of Computer Science
Brunel University London
United Kingdom
george.ghinea@brunel.ac.uk

ABSTRACT

It is of paramount importance that formative feedback is meaningful in order to drive student learning. Achieving this, however, relies upon a clear and constructively aligned model of quality being applied consistently across submissions. This poster presentation raises concerns about the inter-rater reliability of code reviews conducted by teaching assistants in the absence of such a model. Five teaching assistants each reviewed 12 purposely selected programs submitted by introductory programming students. An analysis of their reliability revealed that while teaching assistants were self-consistent, they each assessed code quality in different ways. This suggests a need for standard models of program quality, alongside supporting rubrics and other tools, to be used during code reviews to improve the reliability of formative feedback.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education

Keywords

Programming, Code Review, Code Inspection, Grading, Quality, Assessment, Reliability, Agreement, Consistency.

1. INTRODUCTION

Guidance is important when first learning computer programming to help students develop an appreciation for quality. This often consists of feedback provided during code reviews. However, for such feedback to be meaningful, it should be clear, reliable and constructively align with relevant learning objectives (c.f. [2, 4]). This is because conflicting feedback from different teaching assistants could cause confusion. Previous work suggests that reviews by experienced faculty tend to be correlated, but different reasoning is sometimes applied [1]. However, it remains unclear whether those done by teaching assistants are as consistent. Of particular concern is that the reviews may reflect more on the reviewer than on the student (see [3] for detail on the idiosyncratic rater effect).

ITiCSE'15, July 04–08, 2015, Vilnius, Lithuania

http://dx.doi.org/10.1145/2729094.2754844 ...$15.00.

Table 1: Reliability of Assessment ($E(\alpha) \geq .667$)

Measure	Krippendorff's α
Self-Consistency	.841
Agreement Between Teaching Assistants	.607
Agreement with Faculty Assessments	.522

2. FINDINGS

Five experienced teaching assistants ($> 1yr$) reviewed 12 programs selected from first-year undergraduate computing submissions and made holistic assessments of their quality using a 3-point scale (pass, merit, distinction). Minimal instruction was provided to reflect a less formal context. After two weeks, they re-reviewed the programs. On each occasion the programs were presented in a random order and some elements (e.g., identifiers) were transformed. The data were analysed using Krippendorff's alpha.

The results, shown in Table 1, reveal that while the assessments were adequately self-consistent, there was low inter-rater reliability and there was considerable disagreement with ratings provided by a team of faculty. This finding suggests that teaching assistants apply different standards of program quality when conducting code reviews and therefore require support to improve reliability. As such, this study provides a foundation for future work on the development and evaluation of code review processes, models of program quality, as well as rubrics and other tools.

3. REFERENCES

[1] S. Fitzgerald, B. Hanks, R. Lister, R. McCauley, and L. Murphy. What are we thinking when we grade programs? In *SIGCSE '13*, pages 471–476. ACM, 2013.

[2] A. Pears, J. Harland, M. Hamilton, and R. Hadgraft. Four feed-forward principles enhance students' perception of feedback as meaningful. In *LaTiCE '14*, pages 272–277. IEEE, 2014.

[3] S. E. Scullen, M. K. Mount, and M. Goff. Understanding the latent structure of job performance ratings. *Journal of Applied Psychology*, 85(6):956, 2000.

[4] M. Stegeman, E. Barendsen, and S. Smetsers. Towards an empirically validated model for assessment of code quality. In *Koli Calling '14*, pages 99–108. ACM, 2014.