



Code Reviews to Improve Solo Programming

Dr. Glenn L. Jenkins

2011



Introduction

- Programming difficult for students
 - High attrition rates (Hundhausen et al., 2009, Trytten, 2005)
- Research into pedagogical techniques
 - Collaborative learning approaches (Huggins, 2009, Hundhausen et al., 2009, Sheard et al., 2009)
 - Pair programming (McDowell et al., 2002, McDowell et al., 2006, Williams and Upchurch, 2001)
- Common Problems
 - Pair/Group breakups and their effect on retention (Jacobson and Schaefer, 2008),
 - Contribution of individual students to joint assignments (McDowell et al., 2006)
 - The effect on progression (Simon and Hanks, 2008)



Code Reviews in Pedagogy

- Software Inspections / Peer Code Reviews
 - Are “reviews whose objective is program defect detection” (Sommerville, 2007b).
 - Provide a static test of the code
- Use in Pedagogy
 - Peer Code reviews
 - Trytten (2005) - peer code review
 - Wang et al. (2008) - a loose implementation of formal inspections
 - Hundhausen *et al* (2009) - pedagogical code reviews
 - Individual code reviews
 - Turner et al. (2008) - individual code reviews
 - Humphrey - code reviews as part of the Personal Software Process or PSP (Humphrey, 1997)



Methodology

- Introduction to Programming
 - Year 1 (BSc/HND) Software Engineering and (BSc) Games Development
 - Assignment 1 (term 1 - to Christmas) divided into 4 worksheets
- Pair Programming (2 of 4 worksheets)
 - Introduced last year
 - Improvements in retention but not average marks
 - Possibly similar to that observed by McDowell *et al* (2006)
- Individual Code Reviews
 - Introduced this year
 - Marks associated with code review to encourage students (5%)
 - Code review constructed alongside module and part of tutorials



Results

Year	2008	2009	2010
Term 1 Average	61.40	64.12	63.71
Term 1 Std. Dev.	21.88	15.95	11.57
Term 1 Retention	58.33	73.17	88.24

		No Code Review	Code Review
Worksheet 1	Average	54.57	69.07
	No of Students	14	16
Worksheet 3	Average	52.57	64.36
	No of Students	16	7



Results 2

- Summary of Survey Results
 - Improved quality of program (Agree - 75%)
 - Understood content (65%) and terminology (66%)
 - Liked best
 - Provided opportunity to strategically check for errors (41%)
 - Reminded them of things forgotten (33%)
 - Liked least
 - Time required to undertake the process (65%)
 - Understanding terminology (33%)



Results 3

- Summary of Survey Results continued ...
 - Top reason for not undertaking
 - Focused on assignment (47%)
 - Did not understand (36%)
 - Would use again (90%)
 - Consider code reviews useful (95%)



Conclusions and Future Work

- Improved Programming
 - Observed an increase in code quality
 - Some evidence of an increase in retention
 - Sensitivity of Marking Scheme
- Future work
 - Evaluate the application of this technique at the end of the year
 - Standardize and minimise documentation
 - Apply these techniques to a larger sample