

Unpicking the Pair Programming Process

S. Bryant, P. Romero and B du Boulay
IDEAS laboratory, University of Sussex, UK
s.Bryant@sussex.ac.uk

Pair programming is a method of development that involves two programmers working in tandem on the same problem, usually at a single machine. There are two defined roles – the ‘driver’, who has current control of the keyboard and mouse, and the ‘navigator’, who contributes to the task verbally. These roles are often dynamically swapped between the pair during a pair programming session.

Pair programming is becoming more and more popular, both in industry and academia. It has also been formalised as one of the key practices in the Extreme Programming (XP) methodology. A number of studies have considered the costs and benefits of pair programming and found that it improves software quality, increases programmer enjoyment and improves student retention rates. However, the majority of literature suggesting how and why pair programming works remains largely anecdotal.

We are attempting to ‘unpick’ the pair programming process and have performed four, one-week studies of commercial pair programmers ‘in the wild’. It became clear from the offset that this would be a complex, multi-layered process, requiring use of a number of different analysis techniques. We are therefore using both qualitative and quantitative approaches to consider the following questions:

1. What is the role of artefacts and environment within which pair programming is embedded?
2. To what extent are the pair truly collaborating (rather than cooperating)?
3. Do the pair cover the problem domain more thoroughly by working at different levels of abstraction?

Results have been reported for the first two of these questions, and a verbal protocol analysis scheme is currently be refined for use with regards to the third.