

Social and Human Aspects of Software Engineering

Helen Sharp

The people-intensive nature of software engineering has been understood for some time. Curtis et al [1] cite a number of studies in a variety of contexts demonstrating "the substantial impact of behavioural (i.e. human and organisational) factors on software productivity", while [2] believe that software companies are particularly vulnerable to people problems. Despite this recognition, however, McDermid and Bennett [3] argue that the neglect of people factors in software engineering research has had a detrimental impact on progress in software engineering.

There is much to be learned from quantitative empirical studies of software engineering that focus on the code and software process artifacts, but we should not focus on these to the detriment of studies that investigate human and social aspects of software engineering. Such studies may be quantitative or qualitative in nature, and they may be field-based or experimental. For example, experimental studies may seek to investigate the effects of different support tools or other work artifacts on an individual's performance or cognitive workload; field studies may seek to understand better the information flows around and within a team of agile software developers.

Conducting such studies and analyzing the results in a meaningful way is likely to require collaboration with researchers from social and cognitive sciences, but I believe that the insights to be gained are substantial.

In my own work I conduct qualitative studies 'in the wild', working with practitioners and real business problems. This approach has its drawbacks, including issues of repeatability and generalisability. One way to address these drawbacks is to perform multiple studies and look for similarities. Another approach is to use these exploratory studies to identify hypotheses that can then be pursued through more focused experimental studies.

My current work focuses on agile software teams (e.g. [5]) and motivation in software engineering; previous work has focused on software quality management systems, and object-oriented development (e.g. 4). In each case, I have stressed the people and their goals rather than the technology they are using.

References

1. Curtis B, Krasner H, Iscoe N (1988) A Field Study of the Software Design Process for Large System, *Comms of the ACM* 31(11), 1268-1287
2. Horvat R.V., Rozeman I., Gyorkos J. (2000) Managing the Complexity of SPI in Small Companies, *Software Process and Improvement Journal*, vol 5, 45-54.
3. McDermid JA, Bennett KH (1999) Software Engineering Research: A Critical Appraisal. *IEE Proceedings On Software Engineering* 146(4):179-186.
4. Sharp, H. and Robinson, H. (2005) Some social factors of software engineering: the maverick, community and technical practices. In Proc. of the 2005 WS HSSE2005, St. Louis, Missouri, May 16 - 16, 2005,. ACM Press, New York, NY, 1-6.
5. Sharp, H. and Robinson, H. (2004) An ethnographic study of XP practices, *Empirical Software Engineering*, 9(4) 353-375.