ThinkingISsues

Tony Clear

School of Computer and Information Sciences

Auckland University of Technology,

Private Bag 92006, Auckland 1020, New Zealand

Tony.Clear@aut.ac.nz

Software Engineering and the Academy – Uncomfortable Bedfellows?

With software engineering (S.E.) now beginning to emerge as a distinct discipline, through initiatives such as the SWEBOK [1], the growth of undergraduate programmes in the U.S. [2] and collaborative efforts to offer programmes with breadth and depth [3], the challenges for the academy in balancing rigour and relevance can only become more intense.

For some reason the contrast between the academic and the professional perspectives within the software engineering community is quite marked. This in spite of the fact that the academy has a long history of preparing people for practice in their chosen professions, such as the clergy, medicine and the law. The relatively recent trend towards specialisation that has seen faculty defined as "researchers" [4] rather than the earlier and broader notion of "scholar", seems to partly explain the gulf. Differing career progressions in the distinct communities also play their part.

In the academy the monastic model prevails – novices enter the hallowed halls and undergo instruction as apprentice researchers and educators. The PhD qualification commonly serves as the imprimatur that they are now eligible for junior membership of the academic priesthood. The tenure process must then be navigated to secure a position on the bottom rung of the ladder in many Universities, with promotion and other rewards accruing for publications and research funds gained. Within the S.E.profession a less measured tone prevails. Young, iconoclastic, creative and hierarchy averse, practitioners are rarely PhD qualified, and are recognised by their peers on the strength of their abilities and achievements in developing software and delivering high quality, functioning systems. Rewards for these scarce capabilities can be considerable. Somewhat ironically (given the lack of academic credentials), these practitioners nonetheless undertake defacto research in the software engineering field, pushing the boundaries in their profession as they grapple with the pressing problems of the day for which novel and pragmatic solutions must be found.

At last year's IEEE Frontiers in Education Conference William Wulf [5] reflected upon his experience in moving from an initial period in the academy into practice as a professional engineer designing and providing viable products for customers, and how difficult the transition was for him. Reflecting upon how this experience had enriched his teaching upon his return to the academy, he also noted that few engineering educators possessed any experience of engineering practice. This then, is an acknowledged problem for professional engineers who at least have a well defined and structured body of knowledge to teach. For software engineers with a still young, fluid and evolving discipline, where the boundaries are being encountered and extended in the practice field, the problem is far more significant.

Transfer between the academic and practitioner software engineering communities is rendered difficult because of their differing reward systems. Practitioners will only rarely have the obligatory PhD degree, deemed the entry level to the University, and may not even be considered in an academic hiring process.

The role of software engineering as a discipline within the academy is itself challenging, as it sits in an interdisciplinary space between the computer and information sciences, information systems, engineering and project management domains. This means that software engineering may be captured by a single discipline to the cost of the resulting programmes. In their survey of undergraduate S.E. programs in the U.S., Bagert & Ardis observed that "less than half of the programs had courses specifically devoted to Project Management… Quality Assurance…or Software Requirements" [2]. This either implies that these critical S.E. topics are taught in an interleaved manner within other courses, or have been omitted altogether.

Another barrier in the academy can be an inability to afford recognition to the engaged consultancy-based forms of research, whereby the two communities work together. In the traditional linear model of scientific research, discovery first comes up with a new abstract concept. Only after this scientific research finding comes the engineering of a new technology and resulting products. The place of theory, the crucial role of software, and the creativity and degree of innovation involved in the software process may not be apparent. So the distinction between research and development in the software domain is not always well understood by members of scientific review panels, who sit in judgement on research funding bodies, journal editorial boards and academic promotion committees.

Practicing software engineers have little interest in the turf wars and discipline silos of the academy, but they do have an important and complex role, often at the boundaries of knowledge within a demanding profession. Moreover it is a profession in which many of the problems appear endemic, in which the battle scars of experience appear vital to effective practice, and in which the contributions from within the walls of academy have been arguably far fewer than those developed in the practice field itself. If we consider medicine as an analogous profession, have not the medical educators themselves completed clinical practice requirements? Would doctors who had never practiced be regarded as credible professors of clinical medicine? Why do we privilege the doctoral qualification over the practice credentials in the case of our software engineering professors?

So how does the academy both become and remain credible in the software engineering domain? It seems to me that a mixture of strategies must be adopted, with recruitment and professional development policies being critical.

In our school we are lucky enough to have a number of ex-practitioners with masters level qualifications who can credibly teach software engineering related subjects, armed with a repository of war stories and able to distinguish between the perennial and the emerging issues in the field. However our challenge will be to develop and retain this valuable and highly marketable expertise when they are required to simultaneously teach, professionally upskill through higher studies, develop a personal research programme and profile, and remain current with a volatile and complex field – all while being undervalued in the cloistered environment of the academy.

However it is my belief that "retreaded academics" such as these will have the ability to make the real future contributions to the field, given their deep experientially gained knowledge of the issues in developing quality software. Furthermore, their professional skills in commercial research and development provide a sound base upon which the academic research expertise may subsequently be grafted. So the academy will need to adapt its policies to support modern practice based disciplines such as software engineering. Being prepared to recruit ex-practitioners without PhD qualifications, and recognise their value in non traditional ways may be strategies vital to success in teaching a quality software engineering programme.

1. Abran, A., Moore, J., Bourque, R., Dupuis, P. and Tripp, L. (eds.). *Guide to the*

Software Engineering Body of Knowledge. IEEE Computer Society Press, New Jersey, 2001.

2. Bagert, D. and Ardis, M., Software Engineering Baccalaureate Programs in the United States: An Overview. in *33rd ASEE/IEEE Frontiers in Education Conference*, (Boulder, CO, 2003), IEEE, S3C-1 - S3C6.

3. Hislop, G., Mancoridis, S. and Shankar, P., A Collaborative Bachelor's Degree in Software Engineering. in *33rd ASEE/IEEE Frontiers in Education Conference*, (Boulder, CO, 2003), IEEE, S3C-14 - S13C18.

4. Levy-Leblond, J. Two Cultures or None? *The Pantaneto Forum*, *8. [Online]. Available: http://www.pantaneto.co.uk/issue8/levyleblo nd.htm[16 January 2004].*

5. Wulf, W., Keynote Speech - Engineering as a Human Endeavour. in *33rd ASEE/IEEE Frontiers in Education Conference. [Online]. Available: http://fie.engrng.pitt.edu/fie2003/ [8 November 2003].* (Boulder, Colorado, 2003), IEEE.