Editorial

Rory V. O'Connor

School of Computing Dublin City University Glasnevin, Dublin 9, Ireland Fax +353–1–7005442 E-mail: roconnor@computing.dcu.ie

Biographical notes: Dr. Rory V. O'Connor is a Senior Lecturer in Dublin City University and a Senior Researcher with Lero, The Irish Software Engineering Research Centre. He received a PhD in Computer Science from City University (London) and an MSc in Computer Applications from Dublin City University. He has previously held research positions at both the National Centre for Software Engineering and the Centre for Teaching Computing and has also worked as a Software Engineer and Consultant for several Irish and European technology organisations. His research interests are centred on the processes whereby software intensive systems are designed, implemented and managed. He is also interested in technology adoption issues and the process whereby information technology tools and techniques are evaluated and selected in a commercial setting.

Information technology systems are a significant presence in business and society. They are evolving swiftly in power, sophistication and areas of application. Continually evolving Information and Communication Technologies (ICTs) touch nearly every aspect of our contemporary life and are closely intertwined with human practices and social innovations. In the development of software intensive systems, the importance of the role of human aspects is central. All information technology projects are populated by teams of people who have individual characteristics, weaknesses and strengths. Their characteristics may or may not fit well with the roles they are assigned to play, or the interactions that may be expected of them within the organisational environment. The development of information technology systems poses increasing challenges for such teams, who may be distributed, perhaps mobile, have varied skills, often speak varied languages and have to function in diverse organisational environments.

The more the information technology industry matures, the more it is becoming accepted that the people involved in software development deserve more attention than the processes or technologies themselves (Tomayko and Hazzan, 2004). Many authors such as DeMarco and Lister (1999) and Constantine (2001) consider that people are a critical software development issue, and the human dimension can be even more important than the technical. This is reinforced by the view that systems development is difficult not because of the complexity of technical problems, but because of the social interaction between users and system developers as they learn to create, develop and express their ideas and visions (Fischer, 2003).

2 R.V. O'Connor

Companies tend to say "human resources are our main asset". This is especially true in the development of information technology systems. Although the study of people's work is in itself an existing discipline, we need to take a special interest in analysing the effect of a number of human resources factors on information technology systems development. Information technology development is and should be considered a human activity, although is if often viewed just in technical terms. The development of information technology and software systems is a human-oriented field and as such will always have the openness of other design disciplines such as architecture and graphic design, rather than the hard-edged formulaic certainty of downstream engineering.

This special issue focuses on Human Aspects of Information Technology Development. Its aim is to illustrate the richness and complexity of human aspects of information technology development and highlight some of the issues and challenges that arise during the development of information technology based software systems.

The importance of the human element for the success of software development projects has been historically emphasised by many authors and has been considered from a variety of perspectives. A common element which can be found is the notion of 'mentality' in reference to certain turns of mind that individuals, teams and whole organisations exhibit and which can significantly impact IT projects. Koutsoukos explores these mentality aspects of the human software developer who resides at the core of every information technology development project. In particular the paper focuses on the attitudes, beliefs and thinking of both individual software engineers and software development teams and explores the consequences for software development projects. This paper proposes to make certain people characteristics first-class concerns of software development methods. It presents the notion of a 'mentality pattern' as an abstract primitive, which can be used to capture, organise, communicate and reason about recurring human attitudes, beliefs and ways of thinking that can have important consequences for software projects.

Innovation is an inherent part of modern business. Numerous research studies and real-life case studies have attested to the fact that innovation is a risky business. While there are significant pressures on organisations to innovate and come up with new ideas to improve performance and add to shareholder wealth, there are also increased pressures not to make bad investments. Software organisations do not survive for too long unless they can innovate. Extremely fierce competition and rapid innovation have made collaborative innovation among distributed teams of software engineers an industry norm. The fundamental questions to software organisations are how they can leverage their innovative ('radical') engineers and how they can manage these efforts to yield the greatest effect. Desouza et al. present the results of a study into the role of such 'radical' individuals in technology organisations whose risky behaviour in terms of experiments with new methods or technologies bring innovations into the mainstream of the organisation. The authors set forth some suggestions for identifying such risk takers, discuss the management regimes under which such 'radical engineers' operate and conclude that successful organisations use such individuals to help foster inventions and innovations, thus becoming market leaders.

The development of information technology systems is a knowledge-intensive and collaborative activity. Members of software development teams are considered to be 'intellect workers' or 'knowledge workers' who are characterised as individuals with high levels of education and specialist skill, combined with the ability to apply these

Editorial

skills to identify and solve problems. Ye *et al.* discuss different forms of knowledge collaboration in software development, roles that computers can play to support knowledge collaboration and the associated technical and social challenges. They posit that the overall capability of a software project team is determined not only by the sum of the knowledge of individual developers, but also by the collaboration between developers and tools, and the collaboration among developers. Further, they argue for the need to provide in situ and individualised support for knowledge collaboration through the approach of layered support on demand and illustrate this with a system that they have developed.

Many companies in the ICT sector now have a global reach. This move towards globalisation has been prompted not simply to ensure a presence close to different customer bases, but also for economic reasons. Nowadays it is quite common to find software development projects that are distributed across the globe, with team members from North America, Europe and the Far East. A related phenomenon that is occurring is the outsourcing of significant software projects, particularly from US and European organisations, to countries such as India, Malaysia and China. There is increasing evidence reported in the literature confirming the importance of trust in business relationships and in particular the literature on trust suggests that the establishment of trust is of importance in the working relationship as it leads to more open communication. Siakas et al. explore the significance and the challenges of trust in offshore outsourcing relationships, which, to a high degree, rely on virtual team members, transcending time, space and culture. They discuss the results from a field-study concerning synchronous and asynchronous virtual collaboration of software development team members from three countries, and also surveys carried out in a student virtual environment. Their findings reveal that trust is culture-bound and they call for cultural awareness and other special precautions in software outsourcing.

The theme of cultural awareness highlighted by Siakas *et al.* at the end of their paper is advanced by Cater-Steel and Toleman who examine the impact that national culture has on information technology practices, specifically in the adoption of software engineering standards and best practices. Comparing results from a European and Australian study of adoption of software best practices in 16 different countries, they analyse the issues using Hofstede's cultural dimensions to explore the relationship between Europe/Australia and the cultural dimensions. They discuss the efficacy of the concept of 'national culture' in light of their analysis and the implications of the globalisation of software engineering standards and methodologies for various stakeholders. They conclude with the need to reconsider the concept and measurement of national culture and advise against the use of simplistic frameworks such as that espoused by Hofstede. They encourage researchers to explore the concept of national culture with appropriate research methodologies.

The software development process describes the way an organisation and its people develop software products and supporting services. A process defines what steps the individuals and teams should take at each stage of production and provides assistance in making estimates, developing plans and measuring quality. The process and associated activities are often documented as sets of procedures to be followed during development. Within the software community there is a widely held belief that a better software process results in a better software product, thus the creation, implementation and maintenance of 4 R.V. O'Connor

a software development process can have a profound impact on a technology organisation. Coleman and O'Connor present a study of how software process is applied in actual practice in the software industry, focusing on the role of both the Company Founder and the Software Development Manager and there influence on the initial formation of software development process practices. They contend that the background of the founder of the company and the software development manager combine to create a management style which has a significant impact on the software development process in an organisation.

References

- Constantine, L. (2001) *Peopleware Papers: The Notes on the Human Side of Software*, Prentice Hall.
- DeMarco, T. and Lister, T. (1999) *Peopleware: Productive Projects and Teams*, 2nd ed., New York: Dorset House.
- Fischer, G. (2003) 'Desert island: software engineering: a human activity', *Journal Automated Software Engineering*, Vol. 10, No. 2, pp.233–237.

Tomayko, J. and Hazzan, O. (2004) Human Aspects of Software Engineering, Charles River Media.