Understanding Flexible and Distributed Software Development Processes

Pär J Ågerfalk[a,b] and Brian Fitzgerald[a]
[a]*Dept of Computer Science and Information Systems, University of Limerick, Ireland*
[b]*Methodology Exploration Lab, Dept of Informatics (ESI), Örebro University, Sweden*
par.agerfalk@ul.ie, bf@ul.ie

The minitrack on Flexible and Distributed Software Development Processes addresses two important and partially intertwined current themes in software development: process flexibility and globally distributed software development.

The first theme, flexibility, concerns the current trend in software development to move from traditional plan-based approaches, as suggested by mainstream software engineering and information systems research, towards more agile approaches. The quest for flexibility is obviously evident in the recent development of various agile methods, such as XP and SCRUM. Agile software development methods have run counter to the prevailing wisdom in software engineering. However, rather than being anti method, agile approaches operate on the principle of 'just enough method' and have produced some outstanding successes in software development lately. The quest for flexibility is also apparent in the currently increasing interest in striking a balance between the rigour of traditional approaches and the need for adaptation of those to suit particular development situations. Although suitable methods may exist, developers struggle in practice with selecting methods and tailoring these better to suit their needs.

The second theme, distributed development, concerns another current trend, namely the fact that more and more software development takes place in globally-distributed settings. This is perhaps most evident in the many cases of outsourcing of software development to low-cost countries, but is also relevant in the case of, for example, utilizing local expertise to satisfy local demands. Distributed development puts new demands on the software process imposed by increased complexity related to, for example, communication (formal, informal, potential lack of), co-ordination (time zones, social awareness, task-sharing, domain expertise, delays), co-operation (trust, teamness), control (policies, project mgt, power, uncertainty), culture (social, political), and technology and tools (heterogeneous technology, standardization). Successful methods for distributed development are yet to be developed and validated. The main challenges of remote software development seem to lie in three disparate areas: the process dimension (what specific development processes are necessary for distributed software development), the technology dimension (what tools can be used to achieve remote development – teleconferencing, shared repositories, etc), and the people dimension (how to ensure that developers have deep domain knowledge, how to manage cultural differences and trust issues – often across continents).

The perhaps most interesting aspect covered by this minitrack is the integrated study of the two themes. Interestingly, many of the difficulties faced in globally distributed software development are the same issues put to the fore by agile methods. Nonetheless, we have yet to see a coherent body of research that investigates the crossbreed between agility and distribution, such as agile approaches to distributed development and distribution as a means to tackle some of the problems associated with plan-based development. Another related and topical issue is how to manage a distributed software project where some sites have a strong background in agile methods and others in traditional plan-based development.

The minitrack brings together three papers addressing a number of important topics related to flexibility and distribution. Harris, Hevner and Collins set out to investigate the seemingly contradictory need for control and flexibility in software development, arguing that effective flexible software development processes must indeed provide clear control mechanisms. Erickson and Evaristo bring attention to some of the risk factors associated with distributed projects by exploring the interaction between key project risk factors and different distributed development attributes. Finally, based on real-life experience, Silva and Cunha bring attention to some of the obstacles related to the combination of agile practices with formal plan-based processes.

0-7695-2507-5/06/$20.00 (C) 2006 IEEE