Dear Reader,

Welcome to another issue of SPIP. The articles in this issue contain insights and fresh perspectives on the software process and where it is heading. We hear a lot about developers and skills, but not enough about the teams or groups that form the basic development unit we use for software development.

Software development is typically done in teams; however SPIP has little to say about how to guide and lead teams. New challenges such as agile and extreme development further emphasise the role of teams in the successful delivery of useful value to an organisation. Many of the new developments are predicated on good practice principles which are used more often and more rapidly. Improving our performance pre-supposes an understanding of how teams work together to deliver software. The new challenges also require us to consider where we are heading in terms of software development, SPI and many of the new trends.

Exploring the future works best when we recall the past, i.e. when we consider where we have come from. The first article by Watts Humphrey provides an excellent introduction to the area of software process improvement. The author, Watts Humphrey, will be familiar to most of our readers as the ‘father’ of process improvement and the inventor of software maturity models.

His pioneering work in the area of the software process resulted in the development of the first Capability Maturity Model. In his reflective article Watts recounts the history of process improvement and the changes in thinking that should drive it in the future. Software is developed in teams within larger organisational structures. It is not sufficient therefore to define the principles of performance at high levels of capability and maturity without telling developers and teams how to perform at those top levels. Team Software Process describes the performance patterns and offers part of an integrated family of methods that can be used to improve the performance of teams seeking to substantially improve their practice. Over 52 years in the discipline, Watts Humphrey has helped to identify and overcome many of the challenges. His reflections provide a useful way and a motivation for moving forward and focusing within the SPI movement.

Teams need to be able to work together and thus require a supportive culture and infrastructure. The second article, The role of collaborative support to promote participation and commitment in software development teams, explores the role of collaborative technology in supporting and extending SPI. Technology can play a role in encouraging team members to create a new culture around improvement. Indeed, collaborative technology can make processes, and the environment, seem more explicit thereby enhancing involvement, participation and commitment to process improvement. The
case studies suggest improved understanding of the process and greater interaction may lead to enhanced commitment. The early results show the importance of acceptance and agreement within the content of improvement. Groupware technology can thus be used to enable teams to explore their culture and to foster and encourage further improvement efforts.

The behaviour of software teams is often determined by their position within the organisation and the overriding culture. The structure and the dynamics of an organisation can influence the way projects are handled, however this is not properly acknowledged in the software project management literature. The article, the impact of turbulence on the effectiveness and efficiency of software development teams in small organisations explores the use of an agent-based software process simulation model in enhancing the understanding of the socio-technological interactions that revolve around SPI activities. The results suggest that autonomous and concurrent teams would be most effective in large organisations. They also indicate that agility derived from incremental and iterative development practices is particularly applicable in small organisations as it offers a useful counterbalance to change.

Agility is crucial to the responsiveness of organisations as adapting and responding become operational necessity in most organisations. The changes in the business environment call for a greater focus on timeliness and on the constant delivery of value and benefit. The adaptive organisation requires an equally agile and adaptive approach to improvement. SPI agility: How to navigate improvement projects investigates the different tactics required for SPI projects. It is based on the empirical results derived from 18 SPI projects conducted at Ericsson in Gothenburg over a period of four years. The study highlights the role of two types of strategies for SPI improvement, a plan-based structured approach and an adaptive, agile approach. The different approaches need to be applied in the correct context and may also be combined. Agile SPI practices are particularly suitable for modern uncertain environments as they have the potential to sustain benefits in dynamic environments.

The adoption of agile practices offers new challenges for developers. One of the more popular agile approaches is extreme programming which pulls together a number of successful practices. Adoption of XP practices in the industry – A survey attempts to investigate the use of these practices through a combination of survey data and case studies. The results suggest that metaphor is the least used practice in XP projects. Given the need for communication within teams and with the customer, and the impact on activities further downstream this omission is surprising. The lack of onsite customer in a number of the projects also questions the benefit of agile processes and the work of agile teams. Given the lack of clear definitions related to ‘how’ to conduct these practices it will be interesting to observe the behaviour of developers to determine if they employ alternative strategies. Indeed, as pointed out by Humphrey in the first paper in this issue guidance as to how teams can work efficiently is sorely needed. This may also be the case in agile development.

Enterprises operate in a bewildering array of shifting conditions and dynamic environments. In order to achieve their objectives companies employ a variety of strategies including reuse, product lines and software platforms. Evolving strategies for software architecture and reuse focuses on the frameworks required to achieve business objectives. There is no one size fits all uniform strategy. The article looks at 13 companies and their reuse efforts on a variety of products. The resulting framework provides a range of options for identifying software reuse strategies. It also points to the type of team structure required to facilitate each of the options.

Organising developers into teams enables collaboration and cooperation. It relies on the establishment of commitment within existing cultures and structures. The dynamics of teams and how they form and perform remains a challenge for developers and managers. Effective management of software development requires co-ordination of effort. Moreover, given the tendency to adopt agile practices to cope with dynamic and responsive environments there is an ever growing need to understand how teams operate. Communication is essential, especially in time-critical contexts. Further improvement relies on gaining a clearer understanding of teams and the way they deliver value and fit within organisational structures.

Agile development methods have transformed the delivery of software capabilities. They also have huge implications on how maturity is assessed and how teams work together. The articles in this issue combine the most recent research with best practice and empirical results. They also bring together reflection in practice and years of experience in
SPI and development. The articles cover many cutting edge ideas offering insights into the most recent trends in development and improvement. Reflection in practice takes years to mature and evolve. We are also privileged to have some of the leading thinkers and practitioners in the process movement share their experience and predications for the future.

Altogether, we hope that the articles will inform and motivate practitioners and researchers to continue to focus on software processes. We hope that insights and experiences trigger new process ideas and encourage all readers to share their reflections and research in this area. We are always on the lookout for new reports of industrial good practice, challenges, promising research directions and tried and tested solutions which can be shared with other readers. The SPI community can benefit from the richness of the growing dialogue between researchers and practitioners exploring agile as well as traditional approaches. We look forward to hearing from you about your views and experiences.