The capacity to reflect on practice to engage in a process of continuous learning is an integral part of software development. Central to such reflective practice is the integration of theory and practice in a cyclic pattern of experience and the conscious application of that learning experience.

The capacity to reflect on practice to engage in a process of continuous learning is an integral part of software development. Central to such reflective practice is the integration of theory and practice in a cyclic pattern of experience and the conscious application of that learning experience.

Reflective practice is characterized by an inquiry of engaging in comparison, pondering alternatives, taking diverse perspectives, and drawing inferences. Especially in complex and novel situations that call for high situational awareness, reflective processes have been granted a major role.

Reflective practice is used as developmental practices for organizations, groups, and individuals. It’s an important activity in which people reacquire their experience, think about it, mull it over, and evaluate it. It is this working with experience that is important in learning. However, the social and cultural context in which reflection takes place has a powerful influence over what kinds of reflection it is possible to foster and the ways in which this might be done.

The general approach to reflective practice was outlined by Donald Schön in The Reflective Practitioner (1983) as an epistemology of practice conducive to systematic learning, reflection, and change. Schön emphasized a need for consistent systems for ongoing learning at both the individual and the organizational level. Such a learning system relies on the ability of actors within the system to learn and, subsequently, transmit this learning to other actors.

Reflective practice stands in opposition to a view of the software profession comprising a standardized body of knowledge that has to be acquired by novices, which can then be applied to solve predefined problems in practice. In contrast, a reflective practitioner is characterized by the ability and willingness to question routinized ways of thinking and acting, either after having acted (reflection-on-action) or in the midst of acting (reflection-in-action). The latter makes it possible to alter one’s current course of action by framing the problem in a new way or by improvising on new ways of solving the problem at hand.

This special issue aims to present a set of high-quality methods, tools, and techniques that support effective individual and group reflection. We invite contributions relating but not limited to

- methods and techniques for individual and group reflection on practice;
- organizational approaches for developing abilities for reflection;
- technology for developing and training reflective skills;
- new models of reflection in practice applied to software engineering;
- exploration of the role of cross-disciplinary methods for reflection on software practice;
- development and critique of technologies for reflection on practice;
- case studies and action research investigating reflection in practice, reasoning, and sense-making; and
- experience reports where reflection was or wasn’t effective at reaching desired improvements.

Questions?
For more information about the focus, contact the guest editors:

- Tore Dybå, Chief Scientist, SINTEF, Norway: tore.dyba@sintef.no
- Neil Maiden, Professor, City University London, UK: n.a.m.maiden@city.ac.uk
- Robert Glass, President, Computing Trends, Australia: rlglass@acm.org

Important dates

- 20 Oct 2013: submission deadline
- 15 Dec 2013: acceptance notification
- 19 Jan 2014: major revisions due
- 15 Feb 2014: final version due
- July/Aug 2014: publication

Submission Guidelines

Manuscripts must not exceed 4,700 words including figures and tables, which count for 200 words each. Submissions in excess of these limits may be rejected without refereeing. The articles we deem within the theme and scope will be peer-reviewed and are subject to editing for magazine style, clarity, organization, and space. We reserve the right to edit the title of all submissions. Be sure to include the name of the theme or special issue for which you are submitting.

Articles should have a practical orientation and be written in a style accessible to practitioners. Overly complex, purely research-oriented or theoretical treatments are not appropriate. Articles should be novel. IEEE Software does not republish material published previously in other venues, including other periodicals and formal conference/workshop proceedings, whether previous publication was in print or electronic form.

Full author guidelines:
www.computer.org/software/author.htm
Submission details: software@computer.org
Submit an article: https://mc.manuscriptcentral.com/sw-cs