CALL FOR PAPERS

Special Issue on Refactoring:
Accelerating Software Change

Modern software is rarely written from scratch. It usually incorporates code from previous systems and is itself reincarnated in other programs. Software also isn’t static; it constantly changes as bugs are fixed and features added. Usually these changes are performed by more than one programmer, and not necessarily by the code’s original authors.

Refactoring supports this highly dynamic software life cycle. Basically, refactoring improves a piece of code’s internal structure without altering its external behavior. You can use it to clean up legacy code, to understand a program, and as a preparation for fixing bugs or adding features. Although any behavior-preserving change can be considered a refactoring, many particularly useful and frequently recurring refactoring operations have been identified and catalogued. Over the past decade, popular development environments have started providing automated support for common refactorings, making refactoring less tedious and error-prone.

So, we solicit submissions for this special issue that focus on the real-world application of research, practical experiences, success stories, and lessons learnt in refactoring. Submissions should focus on one or more of these categories:

- experience applying refactoring tools to industrial code, including rigorous analysis of opportunities and challenges when using them;
- industrial experience (for example, good practices and lessons learned) in implementing or managing refactoring in specific application domains (for example, aerospace, banking, mobile, and embedded systems) or domains not traditionally discussed in the refactoring literature (JavaScript, mobile applications, architecture, and so on);
- research papers describing state-of-the-art processes and tools that enable, support, or improve refactoring, with evidence of their use and impact in industrial settings;
- empirical studies of refactoring “in the field,” addressing one or more human, technical, social, or economic issues through qualitative or quantitative studies; and
- studies of refactoring economics, including estimation and measurement of the size, cost, benefits, time frame, and quality for planning and controlling refactoring in actual organizations into practice.

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

- Emerson Murphy-Hill, emerson@csc.ncsu.edu
- Peter Sommerlad, psummerl@hsr.ch
- Bill Opdyke, opdyke@acm.org
- Don Roberts, don.roberts@gmail.com

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

Manuscripts must not exceed 5,400 words including figures and tables, which count as 250 words each. Submissions exceeding this limit might be rejected without refereeing. The articles we deem within the theme’s 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.

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