Foreword

Welcome to the 4th International Workshop on Managing Technical Debt (MTD 2013), co-located with the 35th International Conference on Software Engineering in San Francisco, California. This is the third year that we are holding this workshop co-located with ICSE.

Delivering complex, large-scale systems faces the ongoing challenge of how to balance rapid deployment with long-term value. From Cunningham’s original description—“not quite right code which we postpone making it right”—people have used the metaphor of technical debt to describe many other kinds of debts or ills of software development, including anything that stands in the way of deploying, selling, or evolving a software system or anything that adds to the friction from which software development endeavors suffer: test debt, people debt, architectural debt, requirement debt, documentation debt, or just an amorphous, all-encompassing software debt.

Consequently, the concept of technical debt in software development has become somewhat diluted lately. Is a new requirement, function, or feature not yet implemented “requirement debt”? Do we call postponing the development of a new function “planning debt”? The metaphor is losing some of its strength on one hand.

On the other hand, the practitioner community has shown increased interest in understanding and managing debt. As the pace of software delivery increases and technology rapidly changes, organizations need better guidance on how to insure the sustainability of their software development effort. This is evidenced by the large amount of discussion of the concept of technical debt in the blogosphere, and in particular in the agile software development arena.

The software engineering community has been engaged in a series of workshops to study this phenomenon and improve the way it is handled. A first workshop on technical debt was held at the Software Engineering Institute in 2010 with the goal of identifying open research questions related to managing technical debt in software. Subsequent workshops have sought to come up with a more in-depth understanding of technical debt, its definitions, characteristics, and its different forms. Discussions have shown that there is an increasing need to formulate a clear research agenda that is well aligned with the industry challenges.

We need theoretical foundations and empirical evidence for analyzing short-term versus long-term goals in large-scale projects. We can offer software engineers a foundation for managing such tradeoffs based on models of their economic impacts. Technical debt succinctly communicates the issues observed in large-scale, long-term projects:

- There is an optimization problem where optimizing for the short-term puts the long-term into economic and technical jeopardy when debt is unmanaged.
- Design shortcuts can give the perception of success until their consequences start slowing projects down.
- Software development decisions, especially architectural ones, need to be actively managed and continuously analyzed quantitatively as they incur cost, value, and debt.

The goal of this fourth workshop is to continue the discussion on technical debt as a part of the research agenda for the software engineering field and bring forward work in progress and ideas from the entire community to collectively vet their validity for the future. To support this goal, submissions were sought describing innovative and significant original research, industrial experience, and future trends. The papers were selected after a peer review by at least three members of the program committee. For this 4th workshop we accepted 11 papers.
out of 20 submissions (55%): 6 full research/experience and 5 short position papers. The accepted papers cover topics such as: models for measuring and optimizing technical debt, real options, tool support, supply chains, managing dependencies, and empirical studies.

Managing technical debt is a broad concern of software engineering that blends research and practice. This can be seen from the program and those involved in the workshop program selection process. To encourage interactive discussion, foster brainstorming and community building the workshop will consist of short presentations from the accepted papers. These short presentations will provide a basis for the participants to investigate further open research questions and challenges in practice. It is for that purpose that the program includes sessions dedicated to open discussion.

We thank our program committee for their thorough reviews of the submissions, which enabled the rigorous selection of papers contributing to a strong program. Your attendance here today shapes the future directions of the research and the conversations on the subject, and we look forward to seeing you all in the workshops and other activities to come.

MTD 2013 Workshop Organizers
- Philippe Kruchten, University of British Columbia, Canada
- Robert Nord, Software Engineering Institute, USA
- Ipek Ozkaya, Software Engineering Institute, USA

MTD 2013 Program Committee
- Barry Boehm, University of Southern California, USA
- Eric Bouwers, SIG and Technical University Delft, Netherlands
- Yuangfang Cai, Drexel University, USA
- Rafael Capilla, Universidad Rey Juan Carlos, Spain
- Jane Cleland-Huang, DePaul University, USA
- Patrick Conroy, University of British Columbia, Canada
- Bill Curtis, CAST, USA
- Hakan Erdogmus, Kalemun Research, Canada
- Neil Ernst, Software Engineering Institute, CMU, USA
- Davide Falessi, Fraunhofer Center for Experimental Software Engineering
- Michael Feathers, Independent, USA
- Israel Gat, Cutter Consortium, USA
- Rick Kazman, University of Hawaii, USA
- Jean-Louis Letouzey, Inspearit, France
- Erin Lim, Aquatic Software, Canada
- John D. McGregor, Clemson University, USA
- Steve McConnell, Construx, USA
- David Morgenthaler, Google, USA
- Don O'Connell, Boeing, USA
- Klaus Schmid, University of Hildesheim, Germany
- Carolyn Seaman, University of Maryland, USA
- Will Snipes, ABB, USA
- Wolfgang Strigel, Canada
- Kevin Sullivan, University of Virginia, USA
- Alexandra Szynkarski, CAST Research Labs, France
- Joost Visser, SIG, The Netherlands
- Liming Zhu, NICTA, Australia