

Preface

Getting development done rapidly often means doing little or no requirements engineering (RE). This workshop provided a forum for researchers and practitioners to discuss in depth possible answers to the question on how RE can be seen as a boon rather than as an obstacle to getting development done on time.

In general, requirements engineering literature has been working with the assumption that a system should be clearly specified before its design and implementation can start. Failing to follow a well-defined requirements process has in the past caused tremendous cost overruns, delays and many project failures. One of the reasons often cited is that the cost and time required for fixing an error increases as development goes on. These experiences combined with the fact that the most critical decisions are usually made during the early development phases support the assumption that the investment of upfront effort will pay off during the later phases of development. This is also the conclusion of the comprehensive CHAOS report of the Standish Group, first published in 1995. A survey described in this report shows that almost half of cancelled projects failed due to a lack of requirements engineering effort and that a similar percentage ascribes good requirements engineering as the main reason for project success.

Nevertheless, there are many companies that still do not practice good requirements engineering. One of the main reasons given for skipping the requirements engineering phase is lack of time. Too often, there is high pressure to deliver something to the customer as soon as possible to keep him/her happy. For this reason, a lot of companies have recently shown great interest in the newly emerged agile methods, such as Extreme Programming. However, it appears that agile approaches overlook requirements engineering as a development phase and solve any problems caused by this lack of upfront effort in the next increment or iteration.

So far, these two camps do not seem to be connected nor do their proponents collaborate closely. For this reason, this workshop was a forum that successfully brought these two groups together. Some of the questions that were addressed include: When should which approach be used? How can both approaches be combined? Although good answers to these questions are likely to be several years away, we feel that this forum got us a step closer to an answer.

These proceedings are a first initiative of collecting knowledge about what we have coined as Time-Constrained Requirements Engineering. The answer we got from the community was excellent. We had 14 submissions out of which we selected 12 for publication. All of them were carefully reviewed by the program committee and several of them were edited according to the comments given by reviewers. We are very glad that we achieve in our first meeting such quality papers. These proceedings will be available both as a book as well as a permanent site on the Internet (<http://www.enel.ucalgary.ca/tcre02>), making it possible that these ideas have a wider audience.

We take the opportunity and thank Sol Greenspan and John Mylopoulos who helped us brainstorm the idea of organizing the workshop. We are also grateful to all the program committee members for their excellent and careful reviews. We would also like to acknowledge the support granted by Capes and Departamento de Informática of PUC-Rio (Julio Leite), the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Alberta Software Engineering Research Consortium (ASERC) (Armin Eberlein).

Armin Eberlein
Julio Cesar Sampaio do Prado Leite