

Review summary of "Impact of the Research Community on the Field of Software Configuration Management: Summary of an Impact Report" by Jacky Estublier, et. al.

This impact report surveys the major areas of SCM, sketching the development of these areas and identifying primary sources of the research that contributed to it. The authors present a historical account of both industrial and academic contributions to the field and speak to the role of technology transfer in both directions. (As such, it makes for a good reference in this area.) Topics discussed include component-based and change set-based versioning, version selection techniques, system modeling and building, merging, workspaces, virtual file systems, process support, distributed development and recent approaches to web enabled SCM. What follows is a short summary of our class's discussion and written summary reviews of the paper, ranging from topical observations to constructive criticism. Professor Jim Whitehead taught this graduate class at UCSC during the fall term of the 2003/2004 school year.

In general the comments on the paper were positive, and many felt that it provided a good introduction to the domain. A couple of students, however, stated that there should have been more of a 'big picture' presented in addition to the provided material, especially with respect to the SCM domains' relationship to other areas such as process improvement. It was noted that the primary innovations in SCM technologies seem largely driven by market pressure. It is unfortunate that this pressure has also had a negative impact on the field where, in some cases, more advanced features are not included in product lines due to the perceived complexity. One example of this is a relatively sophisticated SCM tool, ClearCase, abandoning its custom build tool in favor of make, an aging and feature poor utility.

One contention that several people raised was that, in section 2.2, the authors stated SCM "does not address the issues related to early life cycle phases nor by the last one." This contrasted with both the figure presented on the same page, which identified the use of SCM tools by maintenance teams emerging in the 1985-1990 time frame, and the industrial experience of those in the class, who had personally used SCM tools to manage the artifacts produced during early lifecycle phases. Indeed, it was noted that in projects that use spiral or incremental life cycle models would find much use for SCM systems in the early life cycle phases due to the fact that

requirements and design documents are expected to evolve substantially over the life of the project. Perhaps it was the intention of the authors to state, instead, that SCM systems provide no mechanisms for creating those artifacts? If so, this would not be surprising, as it is not the primary concern of SCM to do so. Some clarification of this section would have been nice.

It was clear to all that read the article that the field of SCM continues to evolve and more research is needed in a number of sub-areas. Among the research areas identified were system modeling, configuration selection, advanced versioning features, data mining techniques and especially better integration with other CASE tools. Returning to the point mentioned above, it would be particularly welcome if SCM tools were better integrated with CASE tools that supported earlier phases of the SE project life cycle. All too often, SCM tools are simply used as repositories for source code and, while this is the original intent of these systems, they would better support modern SE tasks if they were used for more than this. An additional area that was identified was that of refining the internal SCM data models themselves, which are often too complex to be represented very efficiently within the bounds of modern practice. The use of a virtual file system was identified as being crucial to this effort, as it facilitates the decoupling of the file abstraction from its underlying representation in the repository.

In conclusion, this paper was a good introduction to SCM as it presented a wide variety of concepts and mechanisms that are fundamental to the domain. However, the lack of common terminology in the field became obvious to those who had previous experience with one or more tools, and the poor grammar detracted from the readability of the text. This latter point gave some the impression that the document had not been proofread sufficiently. In spite of this shortcoming, the general feeling of the class was that the paper was worthwhile to read and provided a good context for further readings and discussions.

Reference:

J. Estublier, D. Leblang, G. Clemm, R. Conradi, W. Tichy, A. van der Hoek, and D. Wiborg-Weber, "Impact of the Research Community on the Field of Software Configuration Management: Summary of an Impact Report," *ACM Software Engineering Notes*, vol. 27, no. 5 (2002), pp. 31-39.