

four to two, consistent with the discussion above.

## SUMMARY

We have described the key activities necessary for designing and analyzing an experiment in software engineering. We began by explaining how to choose an appropriate research technique to fit the goals of your project. In particular, we taught you how to state your hypothesis and determine how much control you need over the variables involved. If control is not possible, then a formal experiment is not possible; a case study may be a better approach.

Next, we explained the principles of formal experimentation. We listed the six stages of an experiment: conception, design, preparation, execution, analysis and dissemination. The design of an experiment was discussed in some detail. In particular, we pointed out that you must consider the need for replication, randomization and local control in any experiment that you plan to perform. We showed you how you can think of your design in terms of two types of relationships between factors (crossing and nesting), and we described several issues to be considered when selecting an appropriate design.

Once your experimental design was determined, we discussed how to analyze the results. We explained how the distribution of the data can influence the choice of analysis technique, as can the purpose of the experiment and the design considerations.

We hope these articles are useful not only in helping you set up your own experiments, but also in assessing the work of others. There is a profusion of experiments reported in the software engineering literature, many of which suggest that you adopt a particular method, tool or technique. With the analysis suggested here, you should develop a critical eye that will enable you to determine when the reported results are valid and whether the results can be applied to your particular situation or organization.

As I am in the process of moving back to the US, I have asked Barbara Kitchenham of the UK National Computing Centre to take over for the next few installments. She will write a series of articles addressing case studies, much in the same way that we have explored the major issues involved in conducting experiments. Both the experimental and case study work was performed as part of the DESMET project, led by the National Computing Centre and funded by the UK Department of Trade and Industry.

## REFERENCES

- Caulcutt, Roland (1991), *Statistics in Research and Development*, Chapman and Hall, London, England.
- Chatfield, Christopher (1993), *Statistics for Technology*, Chapman and Hall, London, England.
- Kitchenham, Barbara (1992), *DESMET Handbook of Data Collection and Metrickation Book 3: Analysing Software Data*, National Computer Centre, Manchester, England.

# REUSE EMPHASIZED AT NEXT PROCESS WORKSHOP

Barry Boehm

USC Computer Science Dept.  
Los Angeles, CA 90089-0781, USA  
boehm@sunset.usc.edu

The 10th International Software Process Workshop, being held June 17-19, 1996, in Dijon, France, will be emphasizing a software reuse-oriented theme: Process Support of Software Product Lines.

Much of the technology currently available to support the software process has focused on the process of developing and evolving a single software product. Increasingly, organizations are finding advantages in product-line software approaches, involving investments in domain engineering, product line architectures, and rapid applications composition with extensive use of commercial-off-the-shelf (COTS) and other reusable software assets. These new approaches involve significant software process challenges: for example, large-scale software packages (COTS and in-house) often provide so much of an application system's desired functionality that the most effective software approach is for the COTS/reuse capabilities to drive the requirements, rather than the traditional requirements-to-capabilities process model.

Candidate issues to be addressed by the Workshop are Reuse-Sensitive Process Models; Product-Line Oriented Process Models; Reuse of Process Elements; and Effect of Product-Line Considerations on Current Process Technology. Examples of more detailed issues are: How are reuse considerations reflected in models of requirements engineering, architecting, risk management, development, testing, evolution, and process maturity? How are process models affected if the focus is not on a single applications product, but on a product line family of applications? How do these considerations of process support of software product lines affect process representation languages, process enactment support, process management tools, the role of human beings in the software process, and the selection of representative example problems for the software process community to address?

The workshop will consist of intensive discussions of these issues by at most 35 participants, selected on a basis of submitted position papers. Prospective participants should submit a maximum three-page position paper by 5 January 1996, explicitly addressing the workshop theme, and suitable for publication in the proceedings. Papers (7 copies or email in ascii or Postscript format only) should be sent to the address above. A Web page at <http://sunset.usc.edu/Events.html> provides more information on the Workshop.