Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

423

Lionel E. Deimel (Ed.)

Software Engineering Education

SEI Conference 1990
Pittsburgh, Pennsylvania, USA, April 2–3, 1990
Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong

Volume Editor
Lionel E. Deimel
Software Engineering Institute, Carnegie Mellon University
Pittsburgh, PA 15213, USA

Carnegie Mellon University
Software Engineering Institute

CR Subject Classification (1987): D.2, K.3.2

ISBN 0-387-97274-9 Springer-Verlag New York Berlin Heidelberg
ISBN 3-540-97274-9 Springer-Verlag Berlin Heidelberg New York

Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

Preface

This volume is the proceedings of *CSEE '90*, the 4th *SEI Conference on Software Engineering Education*, held in Pittsburgh, Pennsylvania April 2 and 3, 1990. This annual conference is sponsored by the Education Program of the Software Engineering Institute (SEI), a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. The conference provides a forum for discussion of software engineering education and training among members of the academic, industry, and government communities.

The 12 papers chosen for presentation at *CSEE '90* were selected from 28 submitted for consideration. The authors of several papers describing particular graduate programs were asked to participate in a panel on graduate programs in software engineering. Brief descriptions of their programs are also included here, as are position papers by members of a panel on industry-university cooperation and an abstract of the keynote address by Professor David Gries.

Selection of papers was done by a Program Committee from the SEI:

Mark Ardis
Maribeth Carpenter
Gary Ford
Harvey Hallman
James Tomayko

In addition to the above people, the following were referees:

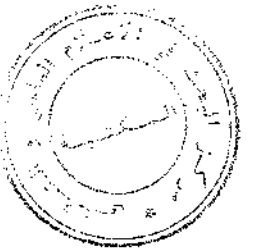
Len Bass, *SEI*
Judy Bamberger, *SEI*
Daniel Berry, *Technion*
Richard Fairley, *George Mason University*
Robert Firth, *SEI*
Gretchen Forbes, *Digital Equipment Corp.*
William Frakes, *Software Productivity Consortium*
Norman Gibbs, *SEI*
John Gilligan, *U.S. Air Force*
Robert Goldberg, *IBM Corp.*
John Knight, *University of Virginia*
Jeffrey Lasky, *Rochester Institute of Technology*
Everald Mills, *Seattle University*
John Musa, *AT&T Bell Laboratories*
George Smith, *Motorola Inc.*
Scott Stevens, *SEI*

Steven Wartik, *Software Productivity Consortium*
Nelson Weiderman, *SEI*
Richard Weis, *University of Hawaii, Hilo*
Robert Winner, *Institute for Defense Analyses*

I would like to express my thanks to the Program Committee and referees, and to Angela Wilkerson and especially Mary Rose Serafini, who helped with administrative details and who kept this whole project on schedule.

Lionel E. Deimel
CSEE '90 Program Chairman

Pittsburgh, Pennsylvania
December 1989



CONTENTS

PANEL DISCUSSIONS:

KEYNOTE ADDRESS:

Instilling Professionalism in Software Engineers

David Gries
Cornell University

Abstract. *I believe it is fair to say that software engineering, as a whole, does not display the same high "professional" attitudes that one finds in other engineering fields. For example, software engineers don't demand (of themselves) the same degree of rigor and clarity in their contracts (specifications) with clients that one finds in other engineering fields. The design, programming, and testing of programs and software systems often take far too long. Programs are usually difficult to understand, leading to the high cost of "maintenance." And few software products bear the guarantee of their authors that the product is correct—in fact, much software comes with a complete disclaimer as to the responsibility of its author!*

The reasons for this situation will be discussed, and some avenues of rectifying it will be explored.

Establishing Motorola-University Relationships: A Software Engineering Training Perspective

George Sanders and George Smith
Motorola Inc.

Abstract. *Motorola requires a skilled software engineering work force to accommodate the growing importance of software within our products and our corporation. The current computer science curricula at most universities do not prepare students to develop industrial strength software and little is available in the way of continuing professional development. Consequently, Motorola has begun a major initiative to develop and deliver software engineering training. This paper discusses Motorola's effort to establish the training needs of its software engineers and the results of that effort. It provides conclusions from the ongoing investigation of other major companies' software engineering training. It contrasts industry-university relationships in the US, Japan, and Europe. Finally, it discusses the establishment of Motorola-university training programs in the United States, Israel, and Japan.*

1. Introduction

Motorola requires, as a corporate policy, that every employee receives a minimum of 40 hours of training per year. This commitment to training is equally expressed in our growing software engineering training program.

Unlike some other organizations, Motorola, with few exceptions, does not have a full time training staff dedicated to software engineering. As the size and scope of the software demand increases, those key software developers who could be called upon to teach become less available because their expertise has simultaneously become more valuable to both their software development project and the Motorola training community. It is

apparent that Motorola-university cooperation in the training of software engineers must play a key role in meeting our software engineering training needs.

2. DACUM

In order to provide the appropriate software engineering training to our engineers, it first became necessary to identify the courses our engineers needed to improve their performance in the workplace.. Unlike general education, industrial training must provide the student with the correct job skills mix that will be used on the job. To begin our curriculum development efforts, Motorola chose to use a curriculum development model called DACUM (Developing a Curriculum) offered by Ohio State University. DACUM is an innovative approach to occupational analysis. It has proven to be a very effective method of quickly determining, at relatively low cost, the tasks that must be performed by persons employed in a given occupational area.

The job model that results from the DACUM analysis is a detailed and graphic portrayal of the duties and tasks involved in the occupation or job studied. Motorola has developed six software engineering models as part of a corporate-wide analysis, including software engineer, senior-level software engineer, software project management, and team leader.

DACUM operates on the following premises:

Expert workers are able to describe/define their jobs better than anyone else.

Any job can be effectively and sufficiently described in terms of the tasks that successful workers in that occupation perform.

Based on this analysis, Motorola concluded that the software engineering process is basically culturally independent. Software engineers across the country as well as around the world perform the same set of tasks. In a training context, this indicated that a set of training courses that are appropriate in one location (e.g.,