PREFACE

The aim of software engineering is to find methods for developing high quality software products at a reasonable cost. As more and more computers are being used in areas in which a malfunction of the system can be a source of serious losses or disturbances to the functioning of the society, the quality of software becomes more and more critical factor of business success, human security and safety. Examples of such application areas are enterprise management, public administration, social insurance or post delivery services. The quality of services offered to the society depends on the quality of software systems that support the functioning of the respective public or private organizations (service providers).

Software engineering consists of a selection of methods and techniques that vary from project to project and evolve in time. The purpose of this volume is to provide an overview of the current work in software development techniques that can help with enhancing the quality of software. The chapters of the volume, organized by key topic area, create an agenda for the IFIP Working Conference on Software Engineering Techniques SET 2006, held October 17–20, 2006 in Warsaw. The seven sections of the volume address the following areas and particular topics:

Software architectures. Methods for structuring the software in order to promote dependability and modifiability, component-based software development, aspect-oriented architectures, distributed and Internet applications.

Modeling. UML-based modeling of component systems, model transformation, semi-formal and formal modeling of software systems using of Petri nets, queuing network models and algebraic calculus.

Project management. Organization-wide process improvement, risk evaluation, modeling and management.

Software Quality. Quality specification and evaluation, user involvement in the quality improvement process, case study.

Analysis and verification methods. Test processes, test automation, test case development and test generation, mutation testing versus aspectoriented response injection, analysis and testing of Java programs, verification of UML state diagrams. headingsddfootoempty venfooteempty

Data management. Knowledge base system engineering, data warehouses and data quality monitoring and maintenance.

Software maintenance. Software refactoring, structuring of Java programs, legacy applications in Web based systems, security problems.

I would like to thank all authors and reviewers who, at the end of the day, create what this is all about.

Krzysztof Sacha *Warsaw, July 2006*