Foreword

“It is not enough to aim; you must hit.” - Italian Proverb

“Software engineering – the “engineering” of software – is part process, part technology, part resource management, and, debatably, until recently, part luck – which make interesting challenges for educators at the undergraduate or graduate level. Learning to be a software engineer – learning about software – learning about engineering (the former, a nebulous topic, the latter an equally nebulous attitude of professionalism) form the target that educators are aiming to hit. Unfortunately, with constant “innovations” in methodologies, technologies, and programming languages, this is a moving target.

“The great aim of education is not knowledge but action.” Herbert Spencer (1820-1903)

Simply put, the aim of this book is to better prepare educators to better prepare students to be better software engineers. The material in the 18 chapters of this book hits the mark by providing proven ammunition for student learning and assessment, curriculum development, innovative teaching methods, and project approaches that solidify classroom concepts, as well as instill an engineering mindset with respect to responsibility, ethics, certification and licensing. It provides a synergistic experience base that can serve the ongoing and future needs of software engineering educators.

“Nothing can add more power to your life than concentrating all your energies on a limited set of targets.” Nido Qubein

To paraphrase Yogi Berra, “Software engineering is 90% aptitude, and the other half attitude.” In my opinion, one of the main challenges facing software engineering educators today is finding a formula for a curriculum that balances theory and application – that channels a student’s aptitude and enhances their ability and capability to be a software engineer. As stated earlier, software is a nebulous topic – not all software applications require the same engineering tradeoffs, but there are key engineering concepts that can be distilled from the experience of others, as captured in the chapters of this book, which will help guide educators in defining and refining software engineering curriculum.

“Aim for success, not perfection. Never give up your right to be wrong, because then you will lose your ability to learn new things and move forward with your life.” Dr. David M. Burns
Perfection is the seductive goal of all software engineering projects - yet perfection has a price that can stand in the way of a successful software solution. The readers of this book will clearly learn new things that I am convinced will lead to success in the classroom that will, in turn, lead to more successful engineering graduates, that will, in turn lead to more successful engineering projects.

In closing, there is one phrase that I first heard used jokingly when I entered the job market only 4 years after the term “Software Engineering” was coined – “Ready, Fire, Aim.” At the time, I did not appreciate its profound applicability to the real world. Software Engineering is the real world. Academia is not, and there lies the challenge that this book addresses. Metaphorically speaking, the material in this book will help educators get ready for software engineering students to learn as well as the educators themselves to teach (by providing a survey of existing learning theories and blended learning approaches as they apply to software engineering education), it will help give educators the ammunition they need to build their software engineering programs and capstone projects (leading to accreditation and more “experienced” students, who can better communicate and work in teams), and finally, it better prepares the students to successfully hit the (moving) target (by giving them an appreciation of ethics and professionalism that they can take outside the classroom).

Will Tracz, PhD
Lockheed Martin Fellow
Editor ACM Software Engineering Notes