

## ANNOUNCING A NEW BOOK

In the College Publications Systems Series

### **Software Engineering in the Systems Context**

Co-Editors: Ivar Jacobson and Harold "Bud" Lawson

INVITATION TO CONTRIBUTE A CHAPTER

***DEADLINE FOR PROPOSALS 31 March 2014***

Since the inception of modern computing the software content of systems has been continually increasing and has often become the dominant element in many complex systems. The need for software capabilities and requirements always come from the wider system context in which the software is an element. Thus the life cycle management of both systems and their software elements has become a central issue. In this volume we seek to portray the considerations that must be taken in relationship to software life cycle management in the context of the system of which they form a part. There are several on-going efforts to address this relationship and we would like to highlight these developments in this book.

After many years of experience with diverse approaches to Software Engineering, the SEMAT (Software Engineering Method and Technology) effort has resulted in an emerging standard by Object Management Group and is presented in the *Essence of Software Engineering* by Jacobson, et.al., (2013). The *Essence* includes a definition of a Kernel that is actionable, extensible and practical. It is more than just a conceptual model and provides:

- A thinking framework for teams to reason about the progress they are making and the health of their endeavors
- A framework for teams to assemble and continuously improve their way of working
- A common ground for improved communication, standardized measurement, and the sharing of best practices
- And, most importantly, a way to help teams to understand where they are and what they should do next

It is clear that most all of the concepts and ideas in respect to Software Engineering as described in the *Essence* can also be applied in Systems Engineering. However, the larger scope of systems must be taken into account in order to meet the need to communicate about the holistic systems context of which the software system is an element. Flood and Carson (1998) have referred to these as the Narrow System of Interest, Wider System of Interest, Environment and Wider Environment. Thus, the software engineer and the

teams he or she participate in will benefit from learning to “think” and “act” in terms of systems in order to be better prepared to analyze the software system in a wider context, make prudent decisions and perform development tasks in an informed manner (see Lawson, 2010).

The quest to exploit commonalities between Software Engineering and Systems Engineering is also prevalent in the international standards arena. After the development of the ISO/IEC 12207 standard on Software Life Cycle Processes in the 1990s, it was determined that a new international standard was required to reflect that fact that software always exists in a systems context. This resulted in the development of ISO/IEC 15288 standard on System Life Cycle Processes. These two standards are in the process of being harmonized where the first step was accomplished by new versions ISO/IEC 12207 and ISO/IEC 15288 (2008). This harmonization is continuing and the fact that software systems and systems in general can be based upon very similar processes continues to evolve.

A main purpose of this book is to explore the similarities and differences between Software Engineering and Systems Engineering as well as the diverse life cycle requirements for the design, development and deployment of software systems in the wider system context. Chapter contributions are solicited that reflect actual on-going efforts to relate Software Engineering to the wider aspects of Systems as well as providing practical experiences, for example, in the form of case studies.

The following are examples of areas of potential contributions:

- Importance of Systems Science and Systems Thinking for Software Engineers
- Affect of Data Science upon Software and Systems Engineering
- What Software Engineers should know about Systems Engineering
- What Systems Engineers should know about Software Engineering
- Applying the SEMAT Kernel and Essence in Software and Systems Engineering
- Achieving Balance between Architecture and Processes, Methods and Tools
- Standards for Software and Systems Engineering
- Methods and Practices of Software Engineering, respective Systems Engineering
- Bodies of Knowledge for Software and Systems Engineering
- Case studies that take account of Software and broader System aspects

The editors invite you to contribute to this book by submitting a short proposal (2-3 pages) consisting of a proposed title, abstract, preliminary outline. Please respond to both editors at the following e-mail addresses:

[ivar@ivarjacobson.com](mailto:ivar@ivarjacobson.com)

[bud@lawson.se](mailto:bud@lawson.se)

References

Jacobson, I, Ng, P.W., McMahon, P. E., Spence, I., and Lidman, S. (2013) *The Essence of Software Engineering: Applying the SEMAT Kernel*, Addison Wesley, Pearson Education.

Flood, R.L. and Carson, E.R. (1998) *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, Second Edition, Penum Press, London and New York.

Harold "Bud" Lawson, *A Journey Through the Systems Landscape*, College Publications, Kings College, London, UK.

ISO/IEC 12207 (2008) *Information technology - Software life cycle processes*, International Standardisation Organisation/International Electrotechnical Commission, 1, rue de Varembe, CH-1211 Geneve 20, Switzerland.

ISO/IEC 15288 (2008) *Systems and software engineering - System life cycle processes*, International Standardization Organization/International Electrotechnical Commission, 1, rue de Varembe, CH-1211 Geneve 20, Switzerland.