Systems Engineering and Software Engineering

> The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

Lead Authors: *Dick Fairley, Tom Hilburn,* ***Contributing Authors:*** *Ray Madachy, Alice Squires*

Software is prominent in most modern systems architectures and is often the primary means for integrating complex system components. Software engineering and systems engineering are not merely related disciplines; they are intimately intertwined. (See Systems Engineering and Other Disciplines.) Good systems engineering is a key factor in enabling good software engineering.

The SEBoK explicitly recognizes and embraces the intertwining between systems engineering and software engineering, as well as defining the relationship between the SEBoK and the Guide to the Software Engineering Body of Knowledge (SWEBOK) (Bourque, and Fairley 2014; IEEE Computer Society 2022).

This knowledge area describes the nature of software, provides an overview of the SWEBOK, describes the concepts that are shared by systems engineers and software engineers, and indicates the similarities and difference in how software engineers and systems engineers apply these concepts and use common terminology. It also describes the nature of the relationships between software engineering and systems engineering and describes some of the methods, models and tools used by software engineers.

☐

Contents

Topics

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The KAs, in turn, are divided into topics. This KA contains the following topics:

- Software Engineering in the Systems Engineering Life Cycle
- The Nature of Software
- An Overview of the SWEBOK Guide
- Key Points a Systems Engineer Needs to Know about Software Engineering
- Software Engineering Features - Models, Methods, Tools, Standards, and Metrics

Discussion

Software engineers, like systems engineers:

- engage in analysis and design, allocation of requirements, oversight of component development, component integration, verification and validation, life cycle sustainment, and system retirement.
- work with or as a component specialist (for example, user interface, database, computation, and communication specialists) who construct or otherwise obtain the needed software components.
- adapt existing components and incorporate components supplied by customers and affiliated organizations.

These commonalities would make it appear that software engineering is merely an application of systems engineering, but this is only a superficial appearance. The differences between the two disciplines arise from

two fundamental concerns:

1. Differences in educational backgrounds (traditional engineering disciplines for SE and the computing disciplines for SWE) and work experiences that result in different approaches to problem solving, and
2. Different ways of applying shared concepts based on the contrasting natures of the software medium and the physical media of traditional engineering.

Table 1 itemizes some of the shared concepts that are applied in different ways by systems engineers and software engineers. Each discipline has made contributions to the other. Table 1 indicates the methods and techniques developed by systems engineers adapted for use by software engineers and, conversely, those that have been adapted for use by systems engineers.

Table 1. Adaptation of Methods Across SE and SWE (Fairley and Willshire 2011) Reprinted with permission of Dick Fairley and Mary Jane Willshire. All other rights are reserved by the copyright owner. *

Systems Engineering Methods Adapted to Software Engineering	Software Engineering Methods Adapted to Systems Engineering
• Stakeholder Analysis • Requirements Engineering • Functional Decomposition • Design Constraints • Architectural Design • Design Criteria • Design Tradeoffs • Interface Specification • Traceability • Configuration Management • Systematic Verification and Validation	• Model-Driven Development • UML-SysML • Use Cases • Object-Oriented Design • Iterative Development • Agile Methods • Continuous Integration • Process Modeling • Process Improvement • Incremental Verification and Validation

The articles in this knowledge area give an overview of software and software engineering aimed at systems engineers. It also provides more details on the relationship between systems and software life cycles and some of the detailed tools used by software engineers. As systems become more dependent on software as a primary means of delivering stakeholder value, the historical distinction between software and

systems engineering may need to be challenged. This is a current area of joint discussion between the two communities which will affect the future knowledge in both SEBoK and SWEBoK.

References

Works Cited

Bourque, P. and Fairley, R.E. (eds.). 2014. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*). Los Alamitos, CA, USA: IEEE Computer Society. Accessed May 25, 2023. Available at http://www.Swebok.org

Fairley, R.E. and Willshire M.J., 2011. *Teaching systems engineering to software engineering students*, *CSEET* 2011, Software Engineering Education and Training, p: 219-226, ISBN: 978-1-4577-0349-2.

IEEE Computer Society. 2022. SWEBOK Guide Version 4.0 beta. Accessed August 28, 2023. Available at https://waseda.app.box.com/s/elnhhnezdycn2q2zp4fe0f2t1fvse5rn.

Primary References

Bourque, P. and R.E. Fairley (eds.). 2014. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Los Alamitos, CA, USA: IEEE Computer Society. Accessed May 25, 2023. Available at http://www.Swebok.org

Brooks, F. 1995. *The Mythical Man-Month,* Anniversary Edition. Boston, MA, USA: Addison Wesley Longman Inc.

Fairley, R.E. 2009. *Managing and Leading Software Projects.* Hoboken, NJ, USA: John Wiley and Sons.

Additional References

Pressman, R. 2009. *Software Engineering: A Practitioner's Approach.* 7th Ed. New York, NY, USA: McGraw Hill.

Schneidewind, N. 2009. *Systems and Software Engineering with Applications*. New York, NY, USA: Institute of Electrical and Electronics Engineers.

Sommerville, I. 2010. *Software Engineering.* 9th Ed.

Boston, MA, USA: Addison Wesley.

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.11, released 25 November 2024