Software Performance Engineering Education: What Topics Should be Covered?

Connie U. Smith
Performance Engineering Services
L&S Computer Technology, Inc. Austin, TX, USA
www.spe-ed.com

ABSTRACT

This presentation considers elements of Software Performance Engineering (SPE) and how they have evolved. It addresses both skills needed by practitioners and areas of research. Which topics should be covered? How can the education cover realistic systems and problems? What is the history of SPE and is it relevant today? Are these topics unique to SPE education? How should SPE education be integrated with other specialties in Computer Science and Engineering?

CCS CONCEPTS

• **Social and professional topics → Computing education programs**; • **Software and its engineering → Software performance**; **Software system models**.

KEYWORDS

Software Performance Engineering (SPE); Performance Modeling; SPE Education; SPE Curriculum

ACM Reference Format:
Connie U. Smith. 2021. Software Performance Engineering Education: What Topics Should be Covered?. In *ACM/SPEC International Conference on Performance Engineering Companion (ICPE '21Companion), April 19–23, 2021, Virtual Event, France*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3447545.3451200

SUMMARY

Software Performance Engineering (SPE) is a software-based, systematic, quantitative approach to constructing software systems that meet performance requirements [7]. SPE prescribes principles for creating responsive software, the data required for evaluation, procedures for obtaining performance specifications, and guidelines for the types of evaluation to be conducted at each development stage. It incorporates models for representing and predicting performance, as well as a set of analysis methods. This discussion focuses on education needs of these software-based aspects of SPE.

Table 1 lists some of the technical skills needed by SPE practitioners. There is a broad range of skills needed including: practical knowledge of software development, performance modeling, benchmarking and performance testing, problem detection and correction of software architecture and design, and performance management and tuning of many types of execution environments.

Table 1: Key Technical Skills for SPE.

Category	Skills
Software Development	Design methods and notations Development paradigms Domain expertise
Performance Modeling & Evaluation	Software models System models Presentation/visualization of results Validation of models vs. systems
Benchmarking & Evaluation	Scientific Method Representative Repeatable
Performance Improvement	Antipatterns and patterns Measurement and tuning

Practitioners also need a variety of practical skills to effectively apply SPE to systems. Practitioners need experience with real systems either as an apprentice on actual systems or applying techniques to a repository of technical details for representative systems. One of the SPE challenges is that often high-risk projects use recent, state-of-the-art technology so SPE professionals may not have experience with it. They must rely on basic problem solving techniques and extra diligence in the modeling, analysis, and validation of findings. It may be helpful to look to the past for technologies with similar characteristics that may apply. Finally, effective performance engineers also have a good background in management skills, communication, team building and problem solving. For example, experience in creating a business case for SPE efforts helps to justify the efforts and get buy-in from management. Effective communication and visualization of performance results helps to get performance improvements implemented.

Business schools make excellent use of case studies based on actual situations [5]. They may be examples of successes or failures and help students learn concepts by studying these realistic situations. They may relay classic concepts as well as recent, new experiences. SPE education would benefit from collecting and providing similar business cases of SPE experiences. They should include both technical aspects and high-level management issues.

ICPE '21Companion, April 19–23, 2021, Virtual Event, France

ACM ISBN 978-1-4503-8331-8/21/04.
https://doi.org/10.1145/3447545.3451200

Winston Churchill is credited with the 1948 quote:

> *Those who fail to learn from history are condemned to repeat it.*

Others have coined similar statements of the same concept perhaps the first being a 1905 version by Santayana, but Churchill's version is often quoted. It is particularly valid for SPE, I have seen the reinvention of technology that was previously well-established.

Performance was the focus of programming in the early days of computing when computer resources were scarce. Systems with critical performance requirements, such as flight control software, were first implemented in an emulator/simulator. Changes were tested in that software before they were allowed in the actual system. The *fix-it-later* philosophy, i.e., do not worry about performance when you build software - you can always fix it later, emerged after the introduction of the IBM360/MVS (and similar systems from other manufacturers). Computer resources became less expensive and better managed by the operating systems; performance problems quickly followed. Commercial, analytical modeling tools emerged in the 1970s. The term *Software Performance Engineering* was coined in 1981 [6].

Thus, we have over 50 years of history in performance engineering. Much of it is not found on the Internet and students and practitioners rarely do literature searches beyond what is available online. In fact the Wikipedia entry for Performance Engineering is incomplete, stale and in need of update [8]. There is no Wikipedia entry for *Software Performance Engineering*. A few authors have presented a partial anecdotal history [1, 3, 4]. We need a diligent effort to compile a scientific collection of dates, facts, sources, etc. on the important contributions to SPE and performance engineering in general.

These topics should be adapted to the teaching environment. For example, which topics represent fundamental skills that are appropriate for high school or university courses and which are appropriate for job training? Do education needs differ for those destined for SPE research versus education versus SPE practice? SPE is constantly evolving, and relevant topics change over time, how does this affect the curriculum? Finally many of the fundamental, practical skills are useful in other domains. Ferrari documented arguments against the insularity of performance evaluation in [2]. Can we find a way to integrate SPE education with other domains? It would be easier to transition to an integrated professional environment.

In summary, this presentation argues for putting the *software* back into performance engineering education. It suggests some of the fundamental technical and practical skills that are needed. Some challenges are:

- teaching about *real* and useful topics, perhaps by developing HBR style business cases.
- covering the history of SPE and performance engineering in general
- integrating with other relevant fields of study in management and engineering.

If we could automate SPE would performance problems disappear and SPE education no longer be necessary?

REFERENCES

[1] T. DeCapua. 2021. https://tinyurl.com/rawxw6au

[2] D. Ferrari. 1986. Considerations on the insularity of performance evaluation. *IEEE Transactions on Software Engineering* SE-12, 6 (1986), 678–683. https://doi.org/10.1109/TSE.1986.6312965

[3] A. Podelko. 2021. https://tinyurl.com/mu9kx2uw

[4] A. Podelko. 2021. https://tinyurl.com/ddcddx7t

[5] Harvard Business Review. 2021. Harvard Business Cases. https://hbsp.harvard.edu/cases/

[6] C.U. Smith. 1981. Increasing Information Systems Productivity by Software Performance Engineering. In *Proc. CMG XII International Conference*. New Orleans, LA.

[7] C.U. Smith and L.G. Williams. 2002. *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*. Addison-Wesley.

[8] Wikipedia. 2021. Performance Engineering. https://tinyurl.com/4cjbvfnc