Mind The Gap: Can And *Should* Software Engineering Data Sharing Become A Path Of Less Resistance?

Extended Abstract*

Ken Wallace†
BAE Systems, Maritime
Portsmouth, UK

ABSTRACT

The facility to process data is, arguably, the defining capability underpinning the transformative power of software: the relationships of each to the other are deep and extensive. This is reflected in the degree to which both software engineering practitioners and researchers rely upon data to direct their endeavours. Ironically however, while both the industrial and research communities are dependent upon data these dependencies present a dichotomy. Practitioners can suffer an abundance of data, much of it dark, which they struggle to interpret and apply beneficially. Isolated by gaps between industry and academia researchers often find themselves lacking data, watching as their industrial counterparts pursue a different and distinct course of action.

Integrating evidence with experience gained in practice and through engagement with research this talk offers an industrial perspective on whether this situation can be improved upon; and what the benefits of achieving this outcome, particularly for practitioners, might be.

CCS CONCEPTS

• **Software and its engineering**; • **Information systems → Data mining**; • **General and reference →** *Measurement*; *Metrics*; *Evaluation*;

KEYWORDS

Software Engineering, Data, Evidence Informed, Industrial Practice, Assessment, Defects, Digital Commons, Evaluation.

ACM Reference Format:
Ken Wallace. 2018. Mind The Gap: Can And *Should* Software Engineering Data Sharing Become A Path Of Less Resistance?: Extended Abstract. In *ICSE-SEIP '18: 40th International Conference on Software Engineering: Software Engineering in Practice Track, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3183519.3183541

*The identified author is solely responsible for the content herein; attribution to BAE Systems is not permitted.

†**ORCID iD** https://orcid.org/0000-0001-7232-0511.

ICSE-SEIP '18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5659-6/18/05.
https://doi.org/10.1145/3183519.3183541

1 DESCRIPTION

As the discipline of software engineering achieves half a century of recognition societal dependence upon software is increasing at an unprecedented rate. With software encroaching ever more deeply into the fabric of the modern world so too are the consequential risks should the resulting reliance upon software prove to be without appropriate foundation. If the discipline of software engineering, as distinct from computer science, is to deliver upon the implicit trust now being placed upon it by society at large there is an increasing need for effective scrutiny and rigour in the conduct of the discipline. This is particularly true in instances where failure of software could have significant adverse consequences [7].

Central to ensuring that software engineering practice is effective and, if necessary, can be demonstrated to be so, is the application of scientific methods to measure, assess and evaluate. Success in doing so is, however, contingent upon having the wherewithall to conduct the requisite empirical studies. From a practitioner's perspective it is also desirable that such investigations yield relevant and beneficial insights that can be applied in practice.

Whereas researchers would argue they have the capability to conduct independent scientific investigation they frequently cite a lack of data as a barrier to doing so. In contrast practitioners, while they may have data, rarely have the time, incentive or even inclination for the conduct of scientific enquiries. Moreover many a practitioner would argue that the insights offered by the research community are insufficient in terms of relevance or benefit, to merit the necessary investment of time and effort on their part [3].

To put the foregoing observations into context the situation identified in respect of the interactions between research and practice is by no means unique to the domain of software engineering. Nevertheless the predominantly commercial nature of software as an enterprise, the relative immaturity of software as an engineering discipline and the pace of innovation and change that software enables all conspire to perpetuate, if not exacerbate, the gaps that exist between practice and research [5]; possibly to a greater extent than many other domains. Indeed this situation contrasts markedly with sectors such as healthcare where society considers it important that legislation, regulation and governance all play significant roles in holding practice to account.

Recognising the existence, and in many ways the entrenched positions of practice and research silos, this talk considers the nature of data sharing in software engineering. Building upon evidence-based software engineering [2, 4, 9] the talk integrates experience of industrial practice [1, 14] together with the application of software engineering research in such practice [13, 15, 16].

Drawing upon evidence from other domains, notably healthcare where evidence-based approaches and data sharing are more widely recognised as being of benefit [6, 8, 12], a hierarchical information model is employed to examine the nature of data sharing in software engineering.

Through decomposition of the titular questions the talk provides an evidence informed assessment of the current state of practice of data sharing in software engineering.

Taking the topic of software defects as an example this assessment is extended to consider whether current models of data sharing can and should continue, given the increasingly social nature of software [10, 11] and the emergence of digital commons. The question of whether these resources can now offer more relevant and effective means to enable the conduct of empirical software engineering research of benefit to practitioners and researchers alike is elaborated upon in the talk.

2 SPEAKER BIOGRAPHY

Ken has been with BAE Systems and antecedent companies since 2001 having previously held engineering and management positions in a number of enterprises working on the development and application of information and software-intensive systems. Prior to this he spent a number of years in academia in the UK and North America working in the field of neuroscience.

Since joining BAE he has continued his involvement in the development of large-scale software-intensive systems in both engineering and management capacities. In additon to acting as the Head of Software Engineering for the Maritime Division, Ken is the founding chair of both the BAE Systems Software Engineering Working Group and the Maritime Software Engineering Council.

Ken retains links with academia having previously acted as an Associate Lecturer in Computing with the Open University (OU), through participation in software engineering research initiatives and as an academic accreditor for the Institution of Engineering and Technology (IET). His research interests include socio-technical aspects of systems and software engineering and range from the application of machine-learning approaches in software engineering through to the development and verification of dependable systems.

A Chartered Engineer, Ken is a member of the ACM, a senior member of the IEEE, a Fellow of the IET and a member of INCOSE. He is a member of the IEEE P7006 Working Group developing a standard for Personal Data Artificial Intelligence Agents and the P7009 Working Group developing a standard for Fail-Safe Design of Autonomous and Semi-Autonomous Systems.

Disclosure

Attendance by the author at ICSE 2018 is funded by BAE Systems.

REFERENCES

[1] Steve Anderson, Paul Sagar, Beth Smith, and Ken Wallace. 2016. What We Have Learnt Adopting Evidence-based Software Engineering for Industrial Practice. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering (EASE '16)*. ACM, New York, NY, USA, Article 30, 1 pages. https://doi.org/10.1145/2915970.2915981

[2] Sarah Beecham, David Bowes, and Klaas-Jan Stol. 2017. Introduction to the {EASE} 2016 special section: Evidence-based software engineering: Past, present, and future. *Information and Software Technology* 89 (2017), 14 – 18. https://doi.org/10.1016/j.infsof.2017.05.002

[3] Sarah. Beecham, Padraig O'Leary, Sean Baker, Ita Richardson, and John Noll. 2014. Making Software Engineering Research Relevant. *Computer* 47, 4 (Apr 2014), 80–83.

[4] Tore Dybå, Barbara.A Kitchenham, and Magne Jorgensen. 2005. Evidence-based software engineering for practitioners. *Software, IEEE* 22, 1 (Jan 2005), 58–65. https://doi.org/10.1109/MS.2005.6

[5] Vahid Garousi, Kai Petersen, and Baris Ozkan. 2016. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Information and Software Technology* 79 (2016), 106 – 127. https://doi.org/10.1016/j.infsof.2016.07.006

[6] Trisha Greenhalgh. 2014. *How to read a paper: The basics of evidence-based medicine*. John Wiley & Sons.

[7] John Hatcliff, Alan Wassyng, Tim Kelly, Cyrille Comar, and Paul Jones. 2014. Certifiably Safe Software-dependent Systems: Challenges and Directions. In *Proceedings of the Future of Software Engineering (FOSE 2014)*. ACM, New York, NY, USA, 182–200. https://doi.org/10.1145/2593882.2593895

[8] Brian Haynes. 2007. Of studies, syntheses, synopses, summaries, and systems: the "5S" evolution of information services for evidence-based healthcare decisions. *Evidence Based Nursing* 10, 1 (2007), 6–7. https://doi.org/10.1136/ebn.10.1.6

[9] Barbara A. Kitchenham, Tore Dybå, and Magne Jørgensen. 2004. Evidence-Based Software Engineering. In *Proceedings of the 26th International Conference on Software Engineering (ICSE '04)*. IEEE Computer Society, Washington, DC, USA, 273–281. http://dl.acm.org/citation.cfm?id=998675.999432

[10] Ingo Scholtes, Markus Strohmaier, and Frank Schweitzer. 2017. Will Computer Science become a Social Science? (2017).

[11] Margaret-Anne Storey, Leif Singer, Brendan Cleary, Fernando Figueira Filho, and Alexey Zagalsky. 2014. The (R) Evolution of Social Media in Software Engineering. In *Proceedings of the on Future of Software Engineering (FOSE 2014)*. ACM, New York, NY, USA, 100–116. https://doi.org/10.1145/2593882.2593887

[12] The PLOS Medicine Editors. 2016. Can Data Sharing Become the Path of Least Resistance? *PLoS Med* 13, 1 (2016), 1–4. https://doi.org/10.1371/journal.pmed.1001949

[13] K. R. Wallace. 2014. I'm A Practitioner Help Me Out Here: Can Evidence Based Software Engineering Really Help Navigate The Iron Triangle Successfully?. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14)*. ACM, New York, NY, USA, 453.

[14] K. R. Wallace. 2014. Safe and secure: re-engineering a software process set for the challenges of the 21st century. In *System Safety and Cyber Security (2014), 9th IET International Conference on*. 1–6. https://doi.org/10.1049/cp.2014.0987

[15] K. R. Wallace. 2015. Can Evidence-Based Software Engineering Contribute To Safer Software?. In *Engineering Systems For Safety: Proceedings of the 23rd Safety Critical Systems Symposium (SSS '15)*. Safety-Critical Systems Club, Bristol, England, UK, 79 – 94.

[16] K. R. Wallace. 2016. Safe Software: A Suitable Case For An Evidence-Based Treatment?. In *Proceedings of the 20th European Safety and Reliability (ESREL '16)*. ESREL, Glasgow, Scotland, UK, 2403 – 2410.