

# Longitudinal Studies in Evidence-Based Software Engineering

Tracy Hall

Longitudinal studies (LS) generate particularly valuable empirical data. There are many reasons for this, most of which are related to the fact that LS are usually large scale. This allows for a range of rich data to be collected. It also means that the scale of data collected should enable statistically significant results to be generated. Furthermore there are also strong temporal aspects to longitudinal studies. These allow changes over time to be tracked which means that the life of a system can be better understood. It also means that the temporal aspects of process change can be identified. The scale and richness of data, collected over the lifetime of a development project, makes for a valuable empirical investigation.

LS can provide more opportunity for contextual data to be collected. Variation in software development environments means that contextual data is particularly important in software engineering. Collecting rigorous contextual data at the right level of granularity means that research findings are more portable. This allows organizations to customize and adapt findings from empirical research and transfer them into their own projects or environments.

LS are rare in software engineering. Researchers find it difficult to access large-scale industrial software development projects over extended periods of time. Such studies are also expensive and time consuming to run. Consequently many empirical studies in software engineering are either short snapshots of industrial projects or else experiments conducted in laboratories isolated from the industrial context. The efficacy of many snapshot empirical studies is compromised by confounding factors. LS allow a sophisticated understanding of software development to emerge. We argue that data collected in such studies can contribute significantly to the maturation of empirical software engineering research.

Extensive use of LS has been made in medicine, for example Remsberg used LS to identify risk factors for developing cardiovascular disease [1]. The usefulness of LS has also been experienced in a few software engineering research studies. Maximilien and Williams [2] conducted a year-long study with an IBM software development group to examine the efficiency of test-driven development.

LS can produce reliable and comprehensive findings that combine technical and social factors in software development. These findings can be presented in a contextualized form that allows them to be more appropriately transferred to other environments. Such findings can directly contribute to the development of bodies of software engineering evidence.

## References

1. Remsberg K, A life span approach to cardiovascular disease risk and aging: The Fels Longitudinal Study. *Mechanisms of Aging & Development*, 124(3), 2003
2. Maximilien, E. M. and Williams, L. 2003. Assessing test-driven development at IBM. In *Proc. of the 25th ICSE*, Portland, Oregon, May 03 - 10, 2003, IEEE Computer Society, Washington, DC, 564-569.