Extending Empirical Studies to Cover More Realistic Industrial Development and Project Management Issues

Marek Leszak

"Real development" issues, observable in many medium size to large size software systems need to be tackled more intensively by empirical research, to increase applicability and concrete transfer of scientific results into practice. This is illustrated by three areas, covering typical problems in today's software project management.

Many predictive and prescriptive models rely on the assumption of steady-state project dynamics, at least for main driving factors. But industrial software development is facing an increasing amount of global development, offshore outsourcing, high turnover rate, requirements volatility, constrained resources (both in interval and in cost), mental project pressure, unforeseen project events, etc..

Granularity of empirical studies is often on overall *system level*. At least for large-scale and complex systems this is often not adequate. If the data would be collected and analyzed on subsystem level (product) or team level (process/project) some interesting insights can be revealed [1,2], not available if the system is studied from a 'black-box' perspective. Characteristics of different subsystems tend often not to be homogenous, causally due to e.g. different team size and expertise, and architectural dependencies. E.g. one of our case studies on defect and file distribution [2] provided evidence that a) per subsystem, delivered defects are a good estimator of in-process defects, and 2) per subsystem, the number of product requirements is highly correlated with in-process defects. In this case study such significance could neither be observed on system-level nor on module-level.

Another often neglected aspect of current empirical research is their scope w.r.t. the lifecycle phases studied. If extended towards overall development lifecycle the research results would widen their scope, leading to intensified and value-added collaboration with industrial software development. E.g. most research results on software reviews and inspections focus on code artifacts, although in real development projects typically a much higher amount of effort is spent on reviewing requirements, design documents, and test plans. It is not surprising to see evidence for review effectivity & efficiency being highly dependent (also) on type of artifact [3].

These exemplified levels of detail, in turn, requires on the industrial organization a high degree of data quality, typically not found in lower maturity organizations. Including more realistic factors from large software systems and real-life projects into empirical studies would improve model adequacy and their predictive power.

References

1. O. Laitenberger, M. Leszak, D. Stoll, K. El-Emam: Quantitative Modeling of Software Reviews in an Industrial Setting. 6th IEEE Int. Symp. on Software Metrics. Nov. 1999
2. M. Leszak, D.E. Perry and D. Stoll: Classification and Evaluation of Defects in a Project Retrospective. Journal of Systems and Software, 61/2002, p. 173-187
3. M. Leszak: Software Defect Analysis of a Multi-Release Telecommunications System.. 6th Int. Conf. on product-focussed software process improvement (PROFES). Oulu, Finland, June 2005

V. Basili et al. (Eds.): Empirical Software Engineering Issues, LNCS 4336, p. 131, 2007.