Model-Driven Engineering and the Impact of a Change

Abstract

Design models describe different viewpoints of a software system—separating functionality, from structure, behavior, or usage. While these models are meant to be separate in their description, they are nonetheless related by manifold dependencies. After all, they describe the same system. Yet, this network of dependencies is also the most significant obstacle to model-driven engineering. It is the root cause for failure to propagate changes correctly and completely. Although change propagation as a whole is a daunting challenge to tackle, this talk suggests an approach for addressing this problem in context of model-driven engineering where incorrect or incomplete changes are detectable in form of the inconsistencies they cause. Understanding the impact of a model changes is thus analogous to the detection and repairing of inconsistencies introduced during these changes.
Keynote Speech 2

Things aren’t always What they Seem: Three Examples of Seemingly Proper Statistical Analyses Leading to Unsubstantiated Software Engineering Claims

Magne Jorgenson

Abstract

Statistical analyses of field data are common in empirical software engineering. Unfortunately, it may be easy to misinterpret field data, when not sufficiently aware of assumptions and limitations of the statistical methods. My keynote illustrates this problem through a critical examination of the statistical analyses of three software engineering claims: i) Larger projects have on average larger percentage cost overrun than smaller ones, ii) There is typically an economy-of-scale in software development, and iii) There is a strong, almost universal, tendency towards over-optimism in the estimation of software development costs. All three claims are seemingly strongly supported by published analyses of software engineering field data. In this keynote I demonstrate how violations of essential analysis assumptions, in particular related to measurement error in the independent variables of regression models and non-random or non-representative sampling, are likely to have affected the analyses. In fact, there are good reasons to believe that the first two claims are unsubstantiated and that the third claim is strongly exaggerated. Without an increased awareness of the limitations and assumptions of statistical analyses, the software engineering community will continue to be in risk of presenting statistical artifacts as underlying software engineering relationships.