

DEPARTAMENTO DE INFORMÁTICA

Experimental Software Engineering

SOFTWARE SYSTEMS / SE.MOVA Team



Miguel Goulão

Assistant Professor / Researcher

PhD in Informatics from FCT/UNL, 2008.
18 years of experience in Experimental Software Engineering and its applications to improve software processes and products.

Objectives

There is an increasing demand for a sound validation of Software Engineering claims and theories which includes support from empirical evidence. An underlying objective in most of my work is to contribute to this shift towards an evidence-based culture, where claims are supported by empirical evidence, following examples from other domains, such as Medicine. My goal is to promote such practices both by advancing the state of the art in Experimental Software Engineering and by applying it to domains like Software Evolution, Requirements Engineering and Software Languages Engineering.

Methodology

By means of experimentation, Experimental Software Engineers seek to validate hypotheses raised by induction (and abduction), aiming at building theories that will allow us to help understand the virtues and limitations of methods, techniques and tools, namely by assessing current Software Engineering claims and theories and by expressing quantitatively the cause-effect relationships among software process characteristics (resources and activities) and software product characteristics. This often involves mining software repositories for collecting data on real-world, large-scale, projects. These techniques can be applied transversally in Software Engineering, contributing to the improvement of the body of knowledge in this discipline.

Expected Results

I am currently applying experimental software engineering techniques to different targets, including:

- Software Evolution prediction models (Fig.1), looking to provide reliable and easy to use models to researchers and practitioners, to improve project management.
- Requirements Engineering models quality evaluation using software metrics to support such evaluation (Fig. 2).
- Software Languages Engineering process improvement by introduction a language evaluation framework in the process (Fig 3).

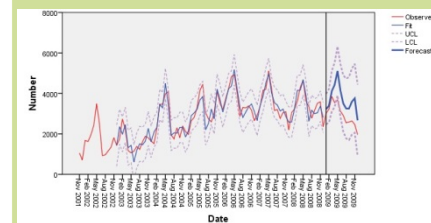


Fig.1 Predicting software change requests

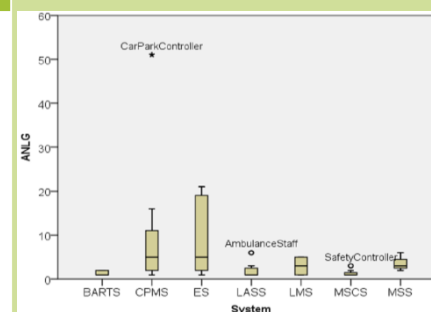


Fig.2 Detecting agents with too many responsibilities

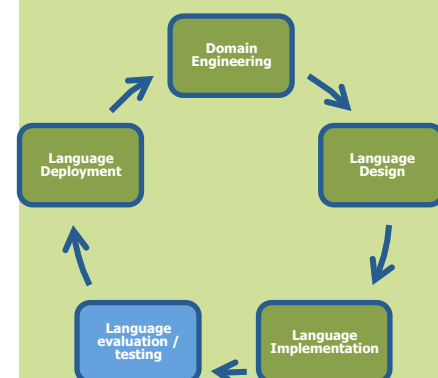


Fig.3 Improving the Software Languages Engineering Process

Funding: