The Relationship between Evolutionary Coupling and Defects in Large Industrial Software (Journal-First Abstract)*

Serkan Kirbas†‡§, Bora Caglayan‡, Tracy Hall§, Steve Counsell§, David Bowes‡, Alper Sen† and Ayse Bener‡
†Bloomberg LP, 3 Queen Victoria Street, London EC4N 4TQ, United Kingdom. skirbas@bloomberg.net
‡Computer Engineering Department, Bogazici University, Istanbul, Turkey. alper.sen@boun.edu.tr
§Data Science Laboratory, Ryerson University, Toronto, Canada. boracaglayan@gmail.com, ayse.bener@ryerson.ca

Abstract—In this study, we investigate the effect of EC on the defect-proneness of large industrial software systems and explain why the effects vary.

I. BACKGROUND
Evolutionary coupling (EC) is defined as the implicit relationship between two or more software artifacts that are frequently changed together. Changing software is widely reported to be defect-prone.

Evolutionary coupling information is generally extracted from the commit history of version control systems. It is based on the assumption that artifacts committed together are logically coupled. This makes EC relatively simple to calculate compared to other types of coupling. For example, structural and semantic coupling are both measured based on the static and text analysis of source code. Often this source code is difficult to obtain from closed source developers. Dynamic coupling analyses execution traces and so requires the software to be executed. EC requires access to only the version control system and is thus a relatively easy way to measure coupling, particularly for industrial closed source systems.

II. METHOD
We analysed two large industrial systems: a legacy financial system and a modern telecommunications system. We collected historical data for 7 years from 5 different software repositories containing 176 thousand files. We applied correlation and regression analysis to explore the relationship between EC and software defects and we analysed defect types, size and process metrics to explain different effects of EC on defects through correlation.

III. RESULTS
The results of our study showed that there was, in general, a relationship between evolutionary coupling and software defects in the industrial software systems under study. We detected a positive correlation between EC measures and defects. Compared to other process measures such as the number of commits and the number of developers, EC measures seem to contain additional, sometimes important, information about defects: for every additional EC, the module is 8% more likely to be defective. However, correlation strength varied across modules and in some modules EC and defects were not correlated. Evolutionary coupling also appeared to be more highly correlated with some types of defects such as code implementation, acceptance criteria and analysis problems. Overall, regression analysis showed that evolutionary coupling may be useful for explaining defects in industrial systems.

IV. CONCLUSION
Although EC measures may be useful to explain defects, the explanatory power of such measures depends on defect types, size and process metrics. Modules which were small in terms of Lines of Code (LOC) and developer numbers tended to be less correlated with evolutionary coupling. Fewer defects due to evolutionary coupling seem to occur in small modules. Interconnections between files in a module can grow quadratically with the number of files. The more inter-related the files are, the more difficult these modules are to understand, change, and correct and thus the more complex the resulting software system. This complexity may eventually lead to defects and this may be one of the reasons for variation across modules. We have also concluded that the dispersion of defect types could be another reason for these varying effects. Different modules have different defect types and evolutionary coupling has different relationships with different defect types. Module characteristics and defect types may also explain why different results are reported by different studies in the literature. Different applications or modules analysed may have different characteristics and defect types.

REFERENCES

*The full version of this work was published as journal article [1].