Software analytics for small software companies: More questions than answers

R. Robbes

University of Chile, Santiago, Chile

CHAPTER OUTLINE

Software analytics have been shown to be useful for many use cases. Examples are numerous, and many of these are mentioned in this book. Software analytics have also been successfully applied in the industry. Again, the examples are numerous. But these examples share one characteristic: they come mostly from large software companies. For instance, Microsoft has a dedicated research group studying its software engineering practice [1], so does Google, or ABB. However, most software companies are small: Richardson and Wangenheim estimated that 85% of software companies had fewer than 50 employees, as of 2007 [2].

These small companies face a different reality than large software companies. They do not have so many resources to allocate to long-term projects, being instead more focused on the short-term. Should software analytics also attempt to address these cases? Should practitioners in these companies take an interest in the topic? If the answers to these questions are far from clear, we bring some elements to the discussion, taking examples from the Chilean software industry as a basis. We particularly focus on the example of Amisoft, a small Chilean software company that successfully used software analytics [3].

THE REALITY FOR SMALL SOFTWARE COMPANIES

Small software companies have different factors that must be taken into account in order to successfully apply software analytics. We identified the following main factors:

- Software projects taken by small software companies may be smaller and shorter than the ones taken by larger companies, limiting the amount of data available.
- The goals and needs of small software companies may be different than the ones in large software companies.
- Resources in small companies are always very tight.

Despite these factors, software analytics could be very important for these companies. Taking the example of Chile, we know that small software companies suffer from a high mortality. Certainly, analytics could be helpful there.

SMALL SOFTWARE COMPANIES PROJECTS: SMALLER AND SHORTER

The Diagnostic Report of the Chilean Software and Services Industry mandated by the GECHS (the Chilean Association of Software and Services Companies) reports on the reality faced by 40 software companies in Chile [4]. Among other findings, companies that were working on projects for clients had project durations of 6 months or less. Service contracts were of similar lengths. Companies developing products did so, in a large majority of cases, in less than a year. A recent survey by the GEMS project found similar results.

The consequences of this is that shorter projects have necessarily much less data to provide for software analytics approaches. If an approach needs at least two years of data to be effective, it will not be applicable in the context of many software companies. Unfortunately, this evaluation criteria is often not present in the evaluation of many Software Analytics or Mining Software Repositories' approaches. Evaluations are generally seen as much more convincing if they involve large-scale projects.

While a few mining software repositories (MSR) approaches (such as the work of Zimmermann et al. [5]) did some evaluation of the technique's performance over time as part of a comprehensive evaluation, it is very uncommon to see this. Further, these approaches did show that a certain amount of time (either measured by amount of time or amount of commits) was necessary for the approaches to reach good performance, telling us that performance with a low amount of data should not be taken for granted, and should definitely be evaluated explicitly. This leads us to formulate this requirement for the design and evaluation of software analytics approaches:

Software Analytics approaches should consider the cases where little data is available, and be evaluated in these cases, so that practitioners in these situations have an idea of the performance they can expect.

DIFFERENT GOALS AND NEEDS

We know of one Chilean small company, Amisoft, that made use of software analytics [3]. Our interview with them was insightful on the differences in priorities between what a large company such as Microsoft needs, and what a small company such as Amisoft needed. We found the overlap between information needs at Amisoft, and the ones at Microsoft [6], to be narrow: of the 17 indicators found to be of interest at Microsoft by Buse and Zimmermann, only three were similar to indicators in use at Amisoft.

Amisoft keeps track of a variety of indicators (for more detail, see [3]), most important of which are how closely the progress follows its expected schedule. Amisoft tracks metrics such as requirement volatility, adherence to its specified process, percentage of time spent on various types of tasks, functional tests and crashes, and client satisfaction. Of note, a significant portion of the data is collected manually (based on employee time sheets) and consolidated manually by project managers and a dedicated data analyst. Amisoft does not collect detailed metrics about the usage of the version control system and the bug tracker (although these tools are in use).

Of course, we cannot answer for all companies based on one datapoint. However, we found that project-level indicators were much more interesting for the CEO than low-level indicators, as could be found in the software repositories that are often mined. To paraphrase the CEO: "I don't really have a need for an approach that tells me where the bugs are. I know that the database access component is the most risky already, hence I personally review changes to it already." Needless to say, actually deploying a defect prediction approach may still have an impact, particularly to reveal other defect-prone modules that are not as prevalent.

However, the CEO particularly appreciated that the approach that Amisoft implemented, based on following higher-level tasks described in the process that they were using, gave him and his project managers far better visibility into their progress. This helped them react much more quickly to delays, and avoid the "heroic" rushes before deadlines that can lead to developer burnout. It also helped them negotiate much more comfortable schedules with their clients, "building in" the estimate for the most likely delays, avoiding that the delay be shown to clients.

As such: *Software analytics for small companies may need to reassess what the most optimal data sources and insights needed are.*

Perhaps a systematic treatment of the issue is needed. The work on information needs of Buse and Zimmermann came a long way toward documenting the needs and perceptions of actual software engineers at Microsoft. A similar study in the context of small software companies could lead to a much more accurate view of what practitioners in small companies need.

WHAT TO DO ABOUT THE DEARTH OF DATA?

We can think of two approaches to address the lack of data for short-term projects. The first is to use finer-grained data: while a single commit in a version control repository gives a single data point, developer interaction data recorded while people use their development tools record hundreds. This can be helpful to paint a higher quality picture of the development sooner, and hence make recommender systems helpful earlier (one of our studies provides evidence of this [7]). Fine-grained interaction data also allows developer to focus on self-improvement via personal feedback. Codealike [8] is a commercial tool geared towards practitioners that collects interaction data from the IDE and the web to provide actionable information to individual developers, and small teams. Codealike uses this data to provide developers dashboards with a variety of information, including analyzing in which activities time was spent, technical debt analysis, and, at the team level, predicting whether an interruption to a given developer will be okay or too disruptive.

The other approach is to use data from other projects, instead of the current project where data might be lacking. Cross-project defect prediction is, for instance, an active area of research, where significant progress has been made to predict defects on projects that have little or no history [9]. Work has been performed in effort estimation as well (which is a topic probably closer to small companies' interests than defect prediction). It remains to be seen how practical these approaches are, and whether other tasks highly relevant to small software companies can be solved in the same way.

WHAT TO DO ON A TIGHT BUDGET?

We'll close this chapter by briefly considering the resources needed to use analytics. Again, as little is known so far, we'll focus on the example of Amisoft. Amisoft's CEO found that the cost of implementing the program was worthwhile in his experience. The upfront cost was relatively high, since a full-time employee (out of 43 at the time) was focused on the project for 3 months, with the CEO himself dedicating significant time (3.5% of the workforce). Once the program was set up, the impact on individual employees was estimated to be 1 h a week of manual data gathering (which could be reduced with more automation), with the full-time employee staying on to consolidate and analyze the data. We also note that this was possible due to the previous effort in the company to formalize the development process: there would not have been anything to measure before that.

In case this appears too high, a lighter-weight entry in analytics would be the individual feedback offered by tools such as Codealike: data collection is entirely automatic, the only thing needed is the discipline to regularly check the dashboards produced by the tool. The only uncertainty is whether insights provided by the tool would align with the insights needed in a small company (they certainly seem to be useful at the developer level).

REFERENCES

[1] Bird C, Murphy B, Nagappan N, Zimmermann T. Empirical software engineering at Microsoft Research. In: Proceedings of CSCW 2010; 2011. p. 143–50.

[2] Richardson I, Von Wangenheim CG. Guest editors' introduction: why are small software organizations different? IEEE Softw 2007;24(1):18–22.

[3] Robbes R, Vidal R, Cecilia Bastarrica M. Are software analytics efforts worthwhile for small companies? The case of Amisoft. IEEE Softw 2013;30(5):46–53, http://users.dcc.uchile.cl/∼rrobbes/p/SOFTWARE-analytics.pdf. Also available online.

[4] Sexto Diagnóstico de la Industria Nacional de Software y Servicios, http://bligoo.com/media/users/0/32814/files/GECHS-6o_informe_diagnostico.pdf; 2008.

[5] Zimmermann T, Weißgerber P, Diehl S, Zeller A. Mining version histories to guide software changes. IEEE Trans Software Eng 2005;31(6):429–45.

[6] Buse RPL, Zimmermann T. Information needs for software development analytics. In: Proceedings of ICSE 2012; 2012. p. 987–96.

[7] Robbes R, Pollet D, Lanza M. Logical coupling based on fine-grained change information. In: Proceedings of WCRE 2008; 2008. p. 42–6.

[8] Codealike. https://codealike.com.

[9] Zhang F, Mockus A, Keivanloo I, Zou Y. Towards building a universal defect prediction model. In: Proceedings of MSR 2014; 2014. p. 182–91.