Are Software Analytics Efforts Worthwhile for Small Companies? The Case of Amisoft

Romain Robbes, University of Chile

René Vidal, Amisoft

María Cecilia Bastarrica, University of Chile

// Amisoft, a Chilean software company with 43 employees, successfully uses software analytics in its projects, but the analytics that are most effective for it differ from those used in larger companies. //

MICROSOFT HAS A dedicated research group in empirical software engineering,[1] and Google employs at least 100 engineers to improve its tools based on analytics (www.infoq.com/presentations/Development-at-Google). Software analytics have seen great industry acceptance—in large companies. However, most companies aren't capable of investing that much in software analytics because the vast majority of them are small. According to Ita Richardson and Christiane Gresse von Wangenheim, 85 percent of software companies have fewer than 50 employees[2]; in Brazil, 70 percent have fewer than 20 employees[3]; in Canada, 78 percent have fewer than 25 employees[4]; and in the US, approximately 94 percent have fewer than 50 employees.[5] Are software analytics viable for small software companies that aren't able to exploit economies of scale, have less manpower to spare, and have less historical information in their software repositories than companies dealing with large software systems such as Google or Microsoft? We decided to explore this in a small company called Amisoft by conducting interviews (see the sidebar "A Note on Methodology").

Amisoft is a 15-year-old software company established in Santiago, Chile. Its main business is developing custom software and maintaining existing systems. Amisoft is also starting to develop off-the-shelf products to complement its service offering. The company has an average of two new development projects a year; however, its seven indefinite maintenance contracts are the projects that bring financial stability. Amisoft has 43 employees; 40 work directly on software maintenance and development. Every employee performs more than one of the traditional software engineering roles in the company (developer, analyst, tester, and so on).

Why Analytics Were Brought to Amisoft

For most of its existence, Amisoft operated under the code-and-fix software development model and encountered traditional issues: delays, cost overruns, poor software quality, and so on. Moreover, Amisoft wasn't growing. Hence, the company undertook an effort to define and formalize a development process—a variant of the Rational Unified

0740-7459/13/$31.00 © 2013 IEEE

Process—which is now used in all its projects. This effort concluded with recent certifications for the International Organization for Standardization and Capability Maturity Models Integration (level 2).

Software analytics came about as a natural consequence of the desire to

- determine whether employees were really following the company's process,
- measure actual adherence to it,
- gather evidence of whether the process was a net positive for the company,
- increase activities' visibility, and
- locate opportunities for improvement.

Amisoft uses software analytics to ensure a reliable schedule for its projects, in three ways:

- make longer-term strategic decisions based on empirical data at the company level;
- make shorter-term tactical decisions during a specific project at the team or individual level; and
- increase the visibility of processes, projects, and tasks, leading to increased employee awareness, faster reaction time, and self-regulating behavior.

Amisoft introduced software analytics first via a three-month pilot study on two projects, and then in the development process of all the other projects in the company. They have been fully instrumented for several months now.

Software Analytics at Amisoft

Amisoft records a range of metrics to measure the overall health of its projects and how well employees adhere to its process. Amisoft derives metrics from several sources, including manual collection.

Data Sources

Amisoft uses several sources of data for its metrics. Employees manually log time sheets, detail how long they spend on each task, and note in which software engineering discipline they're working (such as requirements engineering, analysis and design, technical solution, and verification and validation). Then, Amisoft's metric data analysts inspect the produced artifacts to measure the degree of adherence to the process. Artifacts include requirements documents, source code, meeting minutes, Gantt charts, and status presentations. The full process references more than 90 artifacts.

Requirements are defined on the Project.net application; the data from the tool is regularly collected to follow requirements status. The site also keeps track of changes to the requirements, allowing Amisoft to measure volatility. Functional tests are defined with Testlink (www.teamst.org). Data from Testlink is regularly collected to measure the adherence of the project to specified functional tests and historical trends.

Amisoft measures customer satisfaction via incident and crash reports from clients, new releases of the system to clients, and monthly customer satisfaction surveys, which are delivered at the end of each iteration. The company gathers employee data from internal surveys, personnel changes in projects, courses on specific technologies taken by employees, and so on.

Metrics

From these data sources, Amisoft derives several metrics to assist leaders in creating goals. The most important metrics concern earned value management (EVM), which is based on task and time tracking and is a recognized technique to ensure a project keeps on time and on budget.[6] Amisoft saw this as the most critical aspect because it directly affects clients.

EVM is a set of metrics that compare the planned completion of tasks with the value they deliver (planned value, or PV), the actual tasks that are completed (earned value, or EV), and the effort spent completing the tasks (actual cost, or AC). During project planning, project managers assign each task a value and an estimated time to completion, yielding a predicted, monotonically increasing value curve. Later, the project managers can then compare this curve to

A NOTE ON METHODOLOGY

In gathering the data presented in this article, we tried to minimize our interactions with employees to respect Amisoft employees' schedules. We extracted findings from a two-hour semistructured interview with Amisoft's CEO (who is also the second author of this article). We recorded and archived the interview and later summarized it. We obtained additional information via emails to the CEO, who would in turn contact project managers as needed. We also used data from the company's projects, which we analyzed when we needed more precise information.

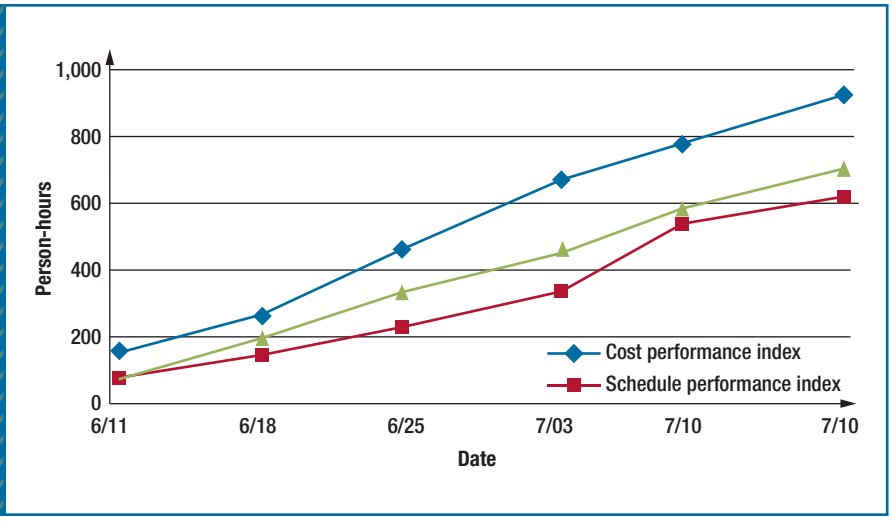

FIGURE 1. Earned value management charts for Amisoft. This project shows delays (the actual earned value is lower than the planned value) and cost overruns (the actual cost is higher than the earned value).

- the actual tasks that are completed to determine if the project is on or behind schedule, and
- the actual effort invested in the tasks—measured in person-hours—to determine if the project is on, over, or under budget.

As an example, the project in Figure 1 shows delays (the actual EV is lower than the PV) and cost overruns (the AC is higher than the EV).

Cost performance (CPI) and schedule performance (SPI) indices are based on EVM, tracking schedules that run both over and under estimates: an index of 1 indicates a project is on time or budget; values over 1 mean the project is ahead of schedule or costs; values under 1 indicate schedule or cost overruns (see Figure 2). The current goal is to stay within 20 percent of the ideal value in both indices at any given time, in effect allowing for schedule overruns of 20 percent per iteration.

Additional metrics. Several metrics in addition to the CPI help evaluate other aspects of Amisoft:

- *Requirement volatility* is the proportion of new, modified, or removed requirements divided by the total number of requirements. There's no hard-set goal other than "reduce requirement volatility" because it has been identified as an issue in the company.
- *Process and schedule adherence* indicates the percentage of artifacts that are correctly produced during project execution. The goal is for 90 percent of the artifacts to be produced correctly and on time. This is critical to ensure that the company doesn't deviate from the process and its plans.
- *Functional tests* are defined for each user requirement. To ensure quality, at the end of each iteration, at least 90 percent of the scheduled functional tests should execute correctly.
- *Software events* monitor occurrences of incident reports by clients, crashes happening in production, and the deployment of new versions in production.
- *Human resources* comprise a registry of changes to the team composition as well as training activities that were taken to address lack of expertise.
- *Qualitative indicators*, such as client satisfaction and employee surveys, use Likert scales.

Amisoft chose the 90 percent goal for tests and adherence metrics because

- it was necessary for better company certifications, and
- most projects already had values over 80 percent, so it was realistic.

Figure 3 shows an anonymized summary of the statuses of all Amisoft projects in a given time period, extracted from a monthly report. From this view, project managers and the general manager can drill down and inspect particular metrics and their evolutions, reacting to deviations from set objectives (see the "Lessons Learned" section for more on the objectives).

Metrics not collected. Amisoft doesn't closely monitor its version control system or defect tracker. The company sees traditional software engineering metrics as a nice thing to have, but such metrics aren't present or are underutilized in the system for cost reasons. In terms of monetary cost and personnel, Amisoft deemed the investment required to deploy such techniques too high in comparison to the expected impact; this led to monitoring business metrics first—their simple instrumentation and interpretation have a larger impact on the clients. Moreover, additional cost needs to be justified to clients, and Amisoft found that clients are hard to convince: often, clients consider internal quality less important than functionality.

Another consideration is that most techniques involving version control and bug-tracking systems work on large datasets. What works for the years of history of a large software system such

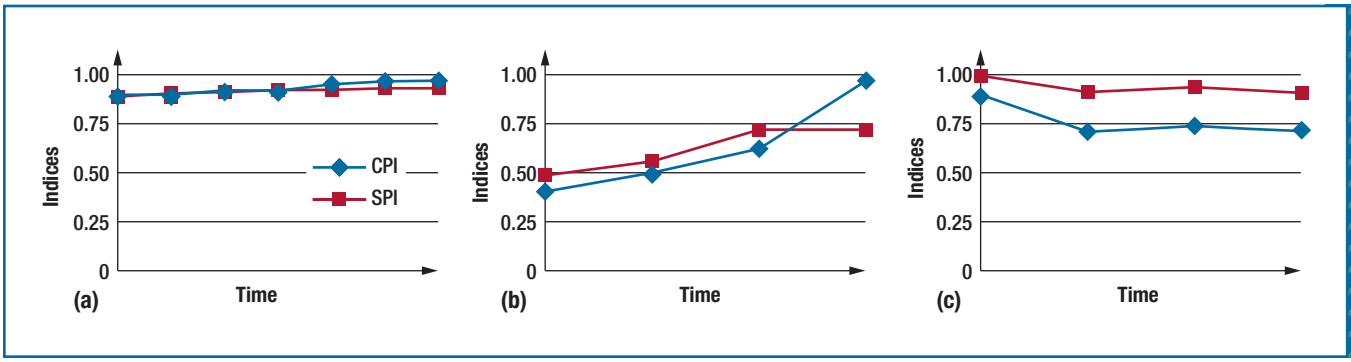

FIGURE 2. Cost and schedule performance charts for Amisoft: (a) the cost and schedule performance indices were maintained or above 1 at the current level during the iteration (33 percent of trends); (b) the indices improved toward 1 (41 percent of trends); and (c) the indices declined below 1 (26 percent of trends).

as Windows might not be effective for a few months of project history like those that come and go at Amisoft. However, Amisoft doesn't exclude these kinds of analyses and plans to focus on improving internal quality in the future.

Data Collection

A key characteristic of Amisoft's data collection is that it's performed weekly, allowing quick reactions to changes and involvement from employees at different stages. Employees fill weekly time sheets, where they declare the tasks they've been working on, the software engineering discipline these tasks belong to, and the time they took to perform each task. Ideally, employees would do this in about 10 minutes each day; however, in actual practice, employees typically fill the log weekly over the course of one hour. Project managers take the data produced by the employees and consolidate it at the project level before giving it to the metric data analyst. The workload for this is 2.5 hours each week for each project manager.

The data analyst then consolidates the data at the company level and integrates all the different data sources as a coherent whole. The analyst updates the metrics every week, issues a company-wide status report every month providing an at-a-glance view of all the projects, and prepares a biannual report that also adds company-level issues (employee surveys, courses, and so on). Because most of the data collection and consolidation process is manual, the data analyst is a full-time employee who spends the majority of his or her time processing the data (around 90 percent of the time) and a minority of the time performing data analysis (concrete analysis within the broader process) and issuing reports.

Benefits of Software Analytics for Amisoft

The complete set of metrics was implemented only recently: it was deployed progressively in all projects, with the oldest being instrumented for eight months at this time of writing. Analytics are already showing benefits; even in a small company, information can be used rapidly. Analytics helped support several strategic and tactical decisions, and increased project progress awareness led to less overwork.

Strategic Decisions

Strategic decisions concern the whole company. Analytics can help those who need to make such decisions.

Scheduling. Based on the historical data and the current goal of admitting cost and schedule overruns of no more than 20 percent, Amisoft adds a cushion factor to project estimations when it bids for contracts. Regardless of the delays that the project experiences, clients will receive software on time; this has obvious benefits in clients' perceptions of the company. Amisoft aims to increase project adherence to the schedule, reducing the cushion to 10 percent, as part of its continuous improvement effort.

Requirement volatility. Measuring requirement volatility helped management realize that it was too high (it was often in double digits during an iteration). The company developed a policy of being careful and methodological in eliciting requirements and required five employees to take additional courses on requirements engineering. The company also informed its clients of the requirement volatility issue, who now participate more willingly in requirements elicitation. Data from previous projects can help emphasize to clients the cost of changing requirements. Although it's too early to assert conclusively that volatility has decreased, measurements in the most recent projects and iterations seem to indicate this is the case.

Verification and validation. Based on the staff-hours invested in each software

engineering discipline, the company realized that the time invested in verification and validation was too low—typically 15 percent of the project's total time—putting software quality at risk. The company decided to increase the amount of resources dedicated to it. However, clients are generally unwilling to invest in verification and validation, so Amisoft is now gathering data to show prospective clients why they should do so. Meanwhile, Amisoft is taking stopgap measures such as assigning idle personnel in a given project to testing activities in other projects.

Tactical Decisions

Project managers use software analytics to make tactical decisions in their projects. It's hard to quantitatively assess the impact of analytics on these decisions because each decision is unique and analytics are part of the evidence leading to a conclusion. However the managers' qualitative perception is that they used them to enact change at various levels in the company. We asked four project managers how they used the indicators.

Personnel. Project managers monitor individual employees in several aspects. One particular project manager made extensive use of the analytics by monitoring two employees who showed regular significant delays to ensure they didn't accumulate further delays; personally revising requirements produced by another employee as requirements elicited by this employee tended to have higher volatility, paying special attention to complex ones; recommending that an employee with a low incident-resolution rate be assisted two hours a day; or simply discussing the situation with the employees as needed.

Client interaction. Two project managers mentioned that they used requirement volatility and incident reports as tools when interacting with clients to emphasize the problems encountered in these areas, prioritize resources toward those efforts, and justify unordinary delays to know if they are correlated with these issues.

Rescheduling. Three project managers reported using planning and task status data to modify task assignments, either to reroute tasks to other people to alter their workload or to change the priority of tasks. One manager mentioned that he uses data from past iterations to schedule the upcoming iterations.

Case Study: Increasing Reactivity to Reduce Work Overload

One characteristic of our data collection process is that most of the metrics are updated weekly. Project managers have used analytics to react to delays (for instance, by rescheduling) and get back on track quickly rather than letting delays accumulate; increased effort is punctual rather than sustained.

> Before Amisoft implemented analytics, delays would often go unnoticed until much later in the iterations.

Given the absence of hard data for the period before the analytics were introduced at Amisoft, we have to rely on anecdotal evidence. Based on the CEO's experience, the situation at Amisoft (once the improved process was introduced) was that most projects were delivered on time but had very high cost in staff-hours and required sustained effort later in the project. Today, the effort is much more evenly distributed but achieves the same results.

To evaluate the reduction in sustained late efforts and the associated burnout, we analyzed the evolution of the CPIs and SPIs of individual iterations to locate rapid adjustments to trends. Iterations usually last between three and six weeks, so weekly metric updates let the team adjust its workload accordingly. We analyzed the data from 29 iterations of five projects and classified each of the resulting 58 metric trends in three categories (see Figure 3).

Furthermore, we looked at the CPI and SPI values at the end of each iteration to determine whether the stated goal of 0.8 or above was reached. This occurred 81 percent of the time; 66 percent of the time, it was above 0.9. This shows that projects react quickly to delays during an iteration. Before Amisoft implemented analytics, delays would often go unnoticed until much later in the iterations, at which point they could have grown to be as large as 50 percent. This would cause considerable risks to the projects, including burnout of employees working long hours or significant delays if a critical employee fell sick at the wrong time. By monitoring the status more often, these situations are much rarer.

The Cost of Software Analytics

There are few drawbacks to such processes, even for a company such as Amisoft.

Initial Cost

The cost of instrumenting the process was important. The first version of the metric-gathering system, a pilot study of two projects (three iterations of

metric definition, metric presentation to the team, metric collection, and metric analysis) took three months to set up. During that time, the CEO spent half of his time on this project and hired a full-time metric analyst to assist him. This amounts to 1.5 out of 43 employees, or 3.5 percent of the workforce. This doesn't account for the cost of formalizing and setting up the development process at Amisoft (which had already been a two-year project).

Running Cost

Another issue is the cost of the process instrumentation. One full-time employee collects the metrics and generates reports. Each employee spends one hour a week (2.2 percent of a 44-hour work week for 37 employees) collecting time-sheet information, and managers spend a significant proportion of their time consolidating information from individual employees to the project level (2.5 hours a week, or 5.7 percent of their work time, for five managers). Including the full-time data analyst (one out of 43 employees), collecting data for software analytics takes approximately 5 percent of the workforce's time.

Resistance

The third issue was resistance from employees who were unsure of how these metrics would be used. They doubted the metrics' usefulness and perceived the manual entry as a significant time loss. It wasn't a time sink (one hour a week), and resistance has since diminished.

Another possible facet of resistance is fake data. However, there's no strong incentive to fake data because managers don't use the metrics to evaluate employees' performances. The analyst, in collaboration with project managers, also checks the metrics when they're integrated and cross-checks them between data sources. So far, the major causes of inconsistencies appear to be clerical errors.

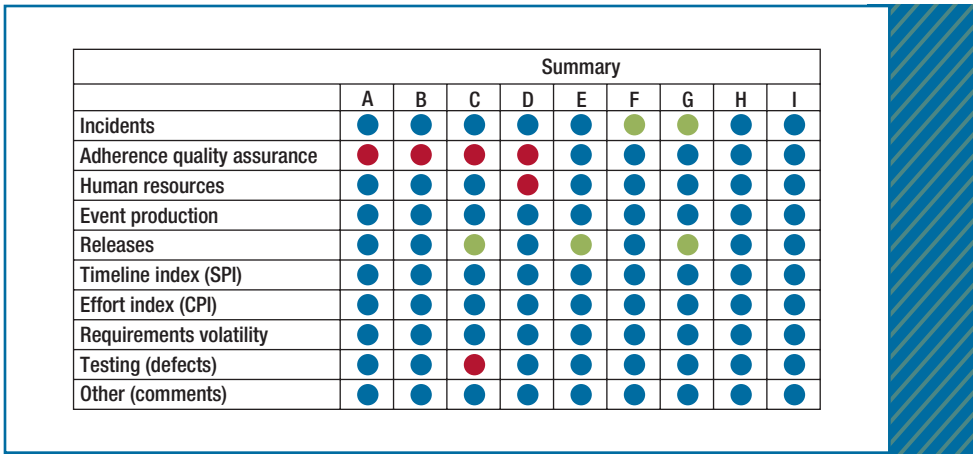

FIGURE 3. High-level status of projects at Amisoft. From this view, project managers and general managers can drill down and inspect particular metrics and their evolutions, reacting to deviations from set objectives.

Lessons Learned

Software analytics are worthwhile—if you follow a process. The main lesson we extracted from this experience is that software analytics are definitely worthwhile, even for a small company like Amisoft. They bring visibility and predictability to the software development process and allow companies to gather evidence in support of a wide range of decisions, from decisions too small to be recorded to long-term changes in company strategies. But data analysis practices lack maturity. Such practices need to be formalized and shared: each project manager used the metrics in a different way. With additional experience and practice sharing, we expect patterns of data analysis to emerge and be consistently adopted by managers. The discovery and consolidation of said patterns should be data analysts' responsibility.

Even smaller companies would also benefit from software analytics. However, the analytics assume that the work is broken down in a series of precise tasks that are assignable, estimable, and trackable. A large proportion of smaller companies—including Amisoft in its early years—don't have the maturity to track tasks precisely; for instance, in small Chilean companies, ad hoc requirements management and losing requirements are still common.[7]

Small companies have different analytics needs. For instance, Microsoft's wide variety of information needs only narrowly overlap with practice at Amisoft.[8] Project managers are interested in higher-level artifacts (requirements at Amisoft, product features at Microsoft, failures, and crashes in both cases), and some of the decision-making scenarios are similar (such as release planning, targeting training, and understanding customers). However, out of 17 indicators at Microsoft, only three (failures, documentation, and engineering activity) are similar to Amisoft's indicators; the others are technical indicators that Amisoft doesn't use yet.

Several factors could contribute to this difference. Microsoft is a much larger company and already has large amounts of data to analyze, which could bias its perceived needs toward questions it can more easily answer. In contrast, Amisoft had to set up data collection from the ground up. Another possible factor is that Amisoft has more direct contact with its clients because

ABOUT THE AUTHORS

ROMAIN ROBBES is an assistant professor at the University of Chile in the Programming Languages and Environments for Intelligent, Adaptable and Distributed Systems research lab. His research interests include empirical software engineering and mining software repositories. Robbes received a PhD in computer science from the University of Lugano. He's a member of IEEE and ACM. Contact him at rrobbes@dcc.uchile.cl.

RENÉ VIDAL is the chief executive officer and founder of Amisoft. His research interests include software process improvement, software metrics, and empirical software engineering. Vidal received an MS in information technology from the University of Chile. Contact him at rene-vidal@amisoft.cl.

MARÍA CECILIA BASTARRICA is an associate professor in the computer science department at the University of Chile. Her research interests include software engineering, software architecture, model-driven engineering, software product lines, and model-driven engineering techniques for modeling software processes. Bastarrica received a PhD in computer science and engineering from the University of Connecticut. Contact her at cecilia@dcc.uchile.cl.

its business model is geared toward specific clients' needs, whereas Microsoft develops mainly off-the-shelf products to be used by many customers.

Amisoft concluded that software analytics are a net positive. Analytics support strategic decisions to address issues that the organization faces, tactical decisions for individual teams, and reactivity. Although the effect of software analytics for longer-term, strategic decisions and punctual, tactical decisions is visible only qualitatively so far, the effect on projects' reactivity—replacing sustained effort at the end of projects and iterations with more punctual effort—is measurable both qualitatively and quantitatively.

For its next steps, Amisoft is increasing its investment in analytics with an automated, integrated system that gathers an extended set of metrics when it's nearing completion. And

Amisoft has only scratched the surface. We believe more actionable information can be discovered because the time dedicated to data analysis is small compared to the time invested in manual collection. Automated processes will free time to perform more advanced analyses.

Acknowledgments

Fondecyt project 11110463 and Program U-Apoya, University of Chile, partly funded Romain Robbes's work. Fondef project D09I1171 ADAPTE partly funded María Cecilia Bastarrica's work.

References

1. C. Bird et al., "Empirical Software Engineering at Microsoft Research," *Proc. Computer Supported Cooperative Work* (CSCW 11), ACM, 2011, pp. 143–150.
2. I. Richardson and C.G. von Wangenheim, "Why Are Small Software Organizations Different?," *IEEE Software*, vol. 24, no. 1, 2007, pp. 18–22.
3. C.G. von Wangenheim, A. Anacleto, and C.F. Salviano, "Helping Small Companies Assess Software Processes," *IEEE Software*, vol. 23, no. 1, 2006, pp. 91–98.
4. C.Y. Laporte, S. Alexandre, and A. Renault, "Developing International Standards for Very Small Enterprises," *Computer*, vol. 41, no. 3, 2008, pp. 98–101.
5. M.E. Fayad, M. Laitinen, and R.P. Ward, "Software Engineering in the Small," *Comm. ACM*, vol. 43, no. 3, 2000, pp. 115–118.
6. *A Guide to the Project Management Body of Knowledge*, Project Management Inst., 2008.
7. A. Quispe et al., "Requirements Engineering Practices in Very Small Software Enterprises: A Diagnostic Study," *Proc. 29th Int'l Conf. Chilean Computer Science Society* (SCCC 10), IEEE CS, 2010, pp. 81–87.
8. R.P.L. Buse and T. Zimmermann, "Information Needs for Software Development Analytics," *Proc. Int'l Conf. Software Eng.* (ICSE 12), ACM, 2012, pp. 987–996.