Are software analytics efforts worthwhile for small companies? The case of Amisoft.

Romain Robbes, PLEIAD Lab, Computer Science Department (DCC), University of Chile, Santiago, Chile

René Vidal, Amisoft Ingenieria Ltda., Santiago, Chile

María Cecilia Bastarrica, MaTE, Computer Science Department (DCC), University of Chile, Santiago, Chile

Abstract—Amisoft, a Chilean software company with 43 employees, successfully uses software analytics in its projects; they support a variety of strategic and tactical decisions, notably resulting in a reduction in overwork of employees. However, the analytics done at Amisoft are very different from the ones used in larger companies.

Keywords: Software Analytics, Case studies

I. Introduction

Software Analytics have seen industry acceptance—in large companies, that is. Microsoft has a dedicated research group in empirical software engineering [1], and Google employs at least 100 engineers improving their tools based on analytics [5].

However, most companies are not capable of investing that much in software analytics. The vast majority of companies are small: according to Richardson and von Wangenheim, 85% of software companies have fewer than 50 employees [8]; in Brazil 70% have fewer than 20 employees [10]; in Canada 78% have fewer than 25 employees [6]; in the USA around 94% have fewer than 50 employees [3].

Are software analytics viable for small software companies, unable to exploit economies of scale, with less manpower to spare, and who have much less historical information in their software repositories than the large amounts of data available in companies dealing with long-lived, large software systems, such as Google or Microsoft?

Amisoft is a 15 year old software company established in Santiago, Chile. Its main business is custom software development and maintenance of existing systems. Amisoft is also starting to develop off-the-shelf products to complement its service offer. The company has on average two new development projects a year; however its seven indefinite maintenance contracts are the projects that bring financial stability.

Amisoft has 43 employees; 40 are directly working in software maintenance and development. Employees perform more than one role in the company, according to the traditional software engineering disciplines (e.g., developer, analyst, tester, etc).

A word on methodology. Given the schedule of Amisoft employees, we minimized our interactions to gather the data presented here. Our findings were extracted from a 2 hour semi-structured interview with Amisoft's CEO (who is also the second author of the paper). The interview was recorded and archived, and later summarized as notes following the flow of the interview. Additional information was obtained by emailing the CEO, who would in turn contact project managers as needed. We also used data from the company's projects that we analyzed when we needed more precise information.

II. Why were analytics 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, etc. Furthermore Amisoft could not grow. Hence the company undertook an effort to define and formalize a development process—a variant of the Rational Unified Process (RUP)— that is now used in all the projects. This effort concluded with recent certifications from ISO and CMMI (level 2).

Software analytics were a natural consequence, to know whether the employees were really following the process, and to measure the actual adherence to it. There was also a necessity to gather evidence whether the process was a net positive for the company, to increase the visibility of the activities, and to locate opportunities for improvement.

These were the reasons to introduce software analytics in the company, first via a 3-month pilot study on two projects, and then in the development process of all the other current projects at Amisoft. They have been fully instrumented for several months now. Amisoft uses software analytics to ensure a reliable schedule for its projects, in three ways:

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

III. Software analytics at Amisoft

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

A. Data sources

Amisoft instruments:

- **Time sheets**, manually filled by employees, detail how long they spend on each task, and by extension in which software engineering discipline (e.g., requirements engineering, analysis and design, technical solution, verification and validation, etc).
- **The artifacts** produced are inspected to measure the degree of adherence to the process. These range from requirements documents, source code, to minutes of meetings, Gantt charts, or status presentations. The full process references more than 90 artifacts.
- **Requirements** are defined in Project.net [1]; the data from the tool is regularly collected to follow requirement status. Project.net keeps track of changes to the requirements, allowing Amisoft to measure requirement volatility.
- **Functional tests** are defined with Testlink[2]. Data from Testlink is regularly collected to measure the adherence of the project to the functional tests that are specified, and historical trends.
- **Client satisfaction** is measured by incident and crash reports made by clients, new releases of the system to the clients, and by monthly customer satisfaction surveys that are delivered at the end of each iteration.
- **Employee data** is gathered from internal surveys, personnel changes in projects, courses on specific technologies taken by employees, etc.

B. Metrics

From these data sources, Amisoft derives the following metrics, and the goals that it sets up with respect to them. The most important metrics have to do with Earned Value Management (EVM). EVM [4], which is based on task and time tracking, was chosen because it is a recognized technique to ensure a project keeps on time and on budget; this was perceived as the most critical aspect, as it impacts clients directly.

EVM is a set of metrics comparing the planned completion of tasks with the value they deliver (**PV, Planned Value**), the actual tasks that are completed (**EV, Earned Value**), and the effort that was spent in completing these tasks (**AC, Actual Cost**).

[1] http://www.project.net/

[2] http://www.teamst.org

During project planning, each task is assigned a value and an estimated time to completion, yielding a predicted monotonically increasing value curve. This curve is compared to the actual tasks that are completed to determine if the project is on schedule or over-schedule, and to the actual effort invested in the tasks—measured in man-hours—to determine if the project is on budget, over-budget, or under-budget. For instance, the project in Figure 1(a) shows delays, since the actual earned value (EV) is lower than the planned value (PV), and cost overruns, since the actual cost (AC) is higher than the earned value (EV).

Based on these metrics, cost-performance (CPI) and schedule-performance (SPI) indices are computed, tracking both over-schedule or under-schedule: an index of 1 indicates a project on-time or on-budget; values over 1 means the project is ahead in terms of schedule or costs; and values under 1 mean cost or schedule overruns (Figure 1b-d). The current goal is to stay within 20% of the ideal value in both indices at any given time, in effect allowing for schedule overruns of 20% per iteration.

Fig. 1. EVM and CPI/SPI charts of at Amisoft

Fig. 2. High-level status of the projects at Amisoft

Additional metrics:

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

The 90% goal for tests and adherence metrics was chosen as it was necessary for better certifications of the company, and because most projects had values over 80%, making the

improvement possible.
Figure 2 shows an anonymized summary of the statuses of all the projects that Amisoft has currently, extracted from the monthly report. From this view, project managers and the general manager can “drill-down” and inspect particular metrics and how they evolved over time, reacting to deviations from the set objectives as shown in Section IV.

Metrics not collected. Amisoft does not monitor its version control system, and does not monitor its defect tracker closely; the “traditional” software engineering metrics are perceived as a nice thing to have but are not present or underutilized in the system for cost reasons. The investment—in terms of monetary cost and personnel—needed to deploy such techniques was deemed too high in comparison to the expected impact, leading to the decision of monitoring business metrics first; their simple instrumentation and interpretation have a larger impact on the clients.

Additional cost needs to be justified, and Amisoft has found that clients are hard to convince: internal quality is not often considered very important by clients and takes a backseat to functionality.

Another factor 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 as Windows may not work for the few months of history of the projects that come and go at Amisoft.

However, Amisoft does not exclude to integrate these kinds of analyses, as it plans to focus on improving internal quality in the future.

C. Data collection

A key characteristic of data collection is that it is performed weekly, allowing quick reactions to changes. Data collection involves all the employees at different stages:

- **Employees** fill weekly time sheets where they declare the tasks they have been working on, the software engineering discipline these tasks belong to, and the time they took to perform each task. This would ideally be done daily, and take on the order of 10 minutes a day, but ends up being filled 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** consolidates the data at the level of the company, and integrates all the different data sources together in a coherent whole. The analyst updates the metrics every week, and issues a company-wide status report every month providing an at-a-glance view of all the projects, and a bi-annual report that also adds company-level issues (employee surveys, capacitations, etc). Since most of the data collection and consolidation process is manual, the data analyst is a full-time employee that spends the majority of his time processing the data (around 90% of the time), and a minority of the time performing data analysis[3] and issuing reports.

IV. Benefits of software analytics for Amisoft

The complete set of metrics has been implemented only recently: it has been deployed in all the projects progressively with the oldest being instrumented for 8 months at this time of writing. Analytics are already showing benefits; even for a small company, information can be used rapidly. We summarize the strategic and tactical decisions that were supported by analytics, and show how increased project progress awareness leads to less overwork.

A. Strategic decisions

- **Scheduling.** Based on the historical data and the current goal of admitting cost and schedule overruns of at most 20%, a cushion factor is added to the estimation of the projects when Amisoft bids for contracts. Regardless of the delays that the project experience, clients will receive software on time, which has obvious benefits in the perception the clients have of the company. Amisoft aims to increase project adherence to the schedule, reducing the cushion to 10%, as part of its continuous improvement effort.
- **Requirement volatility.** Measuring requirement volatility made management realize that it was too high, as it was often in the double digits during an iteration. The company developed a policy of being careful and methodological in eliciting requirements, and made five employees take additional courses on requirement engineering. The clients were also informed, and now participate more willingly in requirement elicitation. Using data from previous projects is helpful to stress the cost of changing requirements to the clients. While it is 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 man-hours invested in each software engineering discipline, the company realized that the time invested in verification and validation was too low—typically 15% of the total time in a project—putting the quality of the software at risk. The company decided to increase in the medium term the amount resources dedicated to it. However, clients are unwilling to invest in verification and validation, hence Amisoft is now gathering data to show prospective clients why they should do so. Meanwhile, stopgap measures are taken, such as assigning idle personnel in a given project to testing activities in other projects.

B. Tactical decisions

[3] Where data analysis refers to the specific step of analyzing concrete data, in the broader process and infrastructure brought by analytics

Project managers use software analytics to make tactical decisions in their projects. We asked four project managers how they used the indicators; we provide a non-exhaustive list of uses:

- **Personnel.** Project managers monitor individual employees in several aspects. One particular project manager made extensive use of this, including: monitoring two employees that showed significant delays on a daily basis to ensure they did not accumulate further delays; personally revising the requirements produced by another employee that tended to have higher volatility in the requirements, paying special attention to complex ones; recommending that an employee with a low incident resolution rate be assisted two hours daily; or simply discussing the situation with the employees when this is 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 towards these efforts; and justify delays out of the ordinary in case they are due to 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 next iterations.

It is hard to quantitatively assess the impact of analytics on these decision, as each decision is unique and analytics are part of the evidence leading to a conclusion. However the qualitative perception of the managers is that they used them to enact change at various levels in the company.

C. Case study: increasing reactivity to reduce work overload

A characteristic of the data collection process is that the majority of the metrics are updated weekly. Analytics have been used to react to delays (for instance by using rescheduling) and get back on track quickly, instead of letting delays accumulate; increased effort is punctual, instead of being sustained.

Given the absence of hard data in the period before the analytics were introduced, 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 at very high human cost and sustained effort later in the project. Nowadays, the effort is much more evenly distributed to achieve the same results.

To evaluate the reduction in sustained late efforts and the associated burnout, we analyzed the evolution of the CPI and SPI indices of individual iterations to locate rapid adjustments to the trends. Iterations last between 3 and 6 weeks usually, so weekly metric updates allow the team to adjust its workload. We analyzed the data from 29 iterations of 5 projects, and classified each of the resulting 58 metric trends in 3 categories:

- 1(b): The indices were maintained at the current level during the iteration, or above 1: 19 out of 58 trends, or 33%.
- 1(c) The indices improved towards 1: 24 out of 58, or 41% of the trends.
- 1(d) The indices declined below 1: 15 out of 58, or 26% of the trends; a minority of cases.

Furthermore, we looked at the value of the CPI and SPI metrics at the end of each iteration, to determine whether the stated goal of 0.8 or above was reached; it was the case 81% of the time (47 out of 58); it was above 0.9 66% of the time. This shows that projects react quickly to delays during an iteration, instead of letting it accumulate. Previously, delays would be noticed much later in the iterations, where they could grow to be as large as 40 or 50%. This would cause considerable risks to the projects, such as burnout of employees working extremely long hours, or significant delays if a critical employee would fall sick at the wrong time. By monitoring the status more often, this situations are much more rare.

To avoid burnout, projects also routinely schedule fewer hours on the projects than they have available on the budget. There are only 2 out of 31 (6.5%) iterations where the hours budgeted are above the available one (effective overtime). Only a third of the iterations schedule more than 90% of the available time on the project. All in all, these figures concord with the anecdotal evidence, yielding strong evidence that Amisoft succeeds in having its employees work balanced hours.

V. THE COST OF SOFTWARE ANALYTICS

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

- **Initial cost.** The cost of instrumenting the process was consequent. The first version of the metric gathering system, a pilot study of two projects, took three months (three iterations of: metric definition, metric presentation to the team, metric collection, and metric analysis) to set up. During that time, the CEO of the company spent half of his time on this project, and a metric analyst was hired full-time to assist him. This amounts to 1.5 out of 43 employees, or 3.5% of the workforce. This does not account for the cost of formalizing and setting up the development process at Amisoft, which was an earlier, two year project.
- **Running cost.** The second issue is the cost of the process instrumentation. There is one full-time employee collecting the metrics and generating reports. Each employee spends one hour a week (2.2% of a 44 hours work week for 37 employees) collecting his time-sheet information, and managers spend a significant proportion of their time consolidating the information from individual employees at the level of projects (2.5 hours a week, 5.7% of their work time, for five managers). Counting the full-time data analyst (one out of 43 employees), collecting data for software analytics takes around 5% of the time of the workforce.
- **Resistance.** The third issue was resistance from the employees, as they were not sure about the usage of these metrics. The usefulness of the metrics was doubted, and the

manual entry was perceived as a significant loss of time. It was shown to not be a time sink (1 hour a week), and resistance has since diminished. Another facet of resistance is the possibility to fake data. There is no strong incentive to do so, as the metrics are not used to evaluate the performance of the employees. The metrics are also checked by the analyst in collaboration with the project managers when they are integrated, and are cross-checked between data sources. So far, the major cause of inconsistencies are clerical errors.

VI. LESSONS LEARNED

Software analytics are worthwhile, if you follow a process. The main lesson extracted from the experience is that software analytics are definitely worthwhile, even for a company the size of Amisoft. They bring visibility and predictability in the software development process, and allow Amisoft to gather evidence in support of a wide range of decisions, from decisions too small to be recorded, up to long-term changes in the strategy of the company. We think that smaller companies would also benefit from it. 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—do not have the maturity to track tasks precisely; for instance, in small Chilean companies, ad-hoc requirement management and losing requirements are still common [7].

Data analysis practices lack maturity. Analysis practices need to be formalized and shared: each project manager used the metrics in a different way. With additional experience and sharing of practices, we expect that "patterns" of data analysis will emerge and be consistently adopted by the managers. The discovery and consolidation of said patterns should be a responsibility of the data analyst.

Small companies have different analytics needs. Buse and Zimmermann surveyed the information needs of Microsoft developers and managers [2]. If there is a wide variety of information needs, the overlap with the practice at Amisoft is narrow. Project managers are interested in higher-level artifacts (e.g., requirements at Amisoft, product features at Microsoft, failures and crashes in both cases), and some of the decision-making scenarios are similar (e.g., release planning, targeting training and understanding customers). However, out of the 17 indicators of Buse and Zimmermann, only 3 (failures, documentation, and engineering activity), are similar to the indicators Amisoft uses; the others are technical indicators that Amisoft does not use yet.

Microsoft is a much larger company, and already has large amounts of data to analyze, which may bias their perceived needs towards questions they can more easily answer. In contrast, Amisoft had to set up data collection from the ground up. Another difference may be that Amisoft has a more direct contact with its clients, as its business model is geared towards the needs of specific clients, while Microsoft develops mainly off-the-shelf products used by large amount of customers.

The study by Treude and Storey [9] of the use of analytics in IBMs Jazz product is another interesting comparison. Most of the individuals studied were developers; they were more interested in indicators related to individual work items, or recent build status, than progress and planning indicators. The study stresses that high-level aspects are the most important for managers, which concurs with our findings.

VII. Conclusion

Software analytics were introduced at Amisoft to gather evidence of the usefulness of software engineering practices. While doing this, Amisoft also concluded that software analytics themselves are a net positive. Analytics support strategic decisions to address issues that the organization faces, tactical decisions for individual teams, and increase reactivity. While the effect of software analytics for longer term, strategic decisions, and punctual, tactical decisions is visible only qualitatively so far, the effect on the reactivity of the projects— replacing sustained effort at the end of projects and iterations with more punctual effort—is measurable both qualitatively and quantitatively.

The analytics at Amisoft are different from the ones in large companies, and in the research community. While they are mostly focusing on technical indicators at the level of source code, versions, and defects, Amisoft focuses first on business- level metrics (schedule, effort, requirements) for their larger potential impact for the client, at lower cost.

Next steps. Amisoft is increasing its investment in analytics, an automated, integrated system that gathers an extended set of metrics is nearing completion. The main roadblock to do it earlier was resources: the project involved two developers full-time for five months, which is a significant investment for Amisoft.

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

VIII. Acknowledgements

The work of Romain Robbes is partly funded by Fondecyt project 11110463, and by Program U-Apoya, University of Chile. The work of Cecilia Bastarrica is partly funded by Fondef project D09I1171 ADAPTE.

References

[1] C. Bird, B. Murphy, N. Nagappan, and T. Zimmermann. Empirical software engineering at microsoft research. In *Proceedings of CSCW 2010*, pages 143–150, 2011.

[2] R. P. L. Buse and T. Zimmermann. Information needs for software development analytics. In *Proceedings of ICSE 2012*, pages 987–996, 2012.

[3] M. E. Fayad, M. Laitinen, and R. P. Ward. Software engineering in the small. *Commun. ACM*, 43(3): 115–118, 2000.

[4] P. M. Institute. *A Guide to the Project Management Body of Knowledge*. Project Management Institute, 2008.

[5] A. Kumar. Qcon talk: Development at the speed and scale of Google, 2010.

[6] C. Y. Laporte, S. Alexandre, and A. Renault. Developing international standards for very small enterprises. *IEEE Computer*, 41(3):98–101, 2008.

[7] A. Quispe, M. Marques, L. Silvestre, S. F. Ochoa, and R. Robbes. Requirements engineering practices in very small software enterprises: A diagnostic study. In *Proceedings of SCCC 2010*, pages 81–87, 2010.

[8] I. Richardson and C. G. von Wangenheim. Guest editors' introduction: Why are small software organizations different? *IEEE Software*, 24(1):18–22, 2007.

[9] C. Treude and M.-A. D. Storey. Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds. In *Proceed- ings of ICSE 2010*, pages 365–374, 2010.

[10] C. G. von Wangenheim, A. Anacleto, and C. F. Salviano. Helping small companies assess software processes. *IEEE Software*, 23(1):91– 98, 2006.

BIOGRAPHIES, PHOTOS, AND CONTACT INFORMATION

Romain Robbes is assistant professor at the University of Chile (Computer Science Department), in the PLEIAD research lab, since 2010. He earned his PhD in 2008 from the University of Lugano, Switzerland and received his Masters degree from the University of Caen, France. His research interests are in Empirical Software Engineering and Mining Software Repositories. He authored more than 50 papers on these topics at top software engineering venues (ICSE, FSE, ASE, EMSE, ECOOP, OOPSLA), and received best paper awards at WCRE 2009 and MSR 2011. He is program co-chair of WCRE 2013, is involved in the organization of several other conferences (ICSE, ICSM, ICPC, IWPSE) and the recipient of a Microsoft SEIF award in 2011. He is a member of the IEEE and the ACM.

Contact information:

Email: rrobbes@dcc.uchile.cl

Romain Robbes

University of Chile

Blanco Encalada 2120, Off. 308

Santiago, Chile

René Vidal is the Chief Executive Officer of the software company Amisoft, which he founded in 1997. In parallel to working at Amisoft, he obtained a Civil Engineering Degree, a Master's Degree in Financial Management and a Master's Degree in Information Technology from the University of Chile, and is pursuing a Master's Degree in Business Administration from the Catholic University of Chile.

Contact Information:

Email: rene-vidal@amisoft.cl
René Vidal,

Amisoft Ltda.

Compañía 1390, Off 2203

Santiago, Chile

María Cecilia Bastarrica is an Associate Professor at the Computer Science Department, Universidad de Chile. She coordinates the MaTE group (Model and Transformation Engineering) since 2007. She received her Ph.D. in Computer Science and Engineering from the University of Connecticut in 2000, a Master of Science from the Catholic University of Chile in 1994, and a Bachelor in Engineering from the Catholic University of Uruguay in 1991. Her main research topics are software engineering, software architecture, model-driven engineering, and software product lines. Lately, her work has focused on applying using MDE techniques for modeling software processes.

Contact information:

Email: cecilia@dcc.uchile.cl

Cecilia Bastarrica

University of Chile

Blanco Encalada 2120, Off. 411

Santiago, Chile