Predicting the health of a project? An assessment in a major IT company

Vincent Blondeau, Sylvain Cresson, Pascal Croisy, Anne Etien, Nicolas Anquetil, Stéphane Ducasse

Predicting the health of a project? An assessment in a major IT company

Vincent Blondeau[1,2] Sylvain Cresson[1]
Pascal Croisy[1]

Anne Etien[2] Nicolas Anquetil[2]
Stéphane Ducasse[2]

[1] Worldline,
Z.I Rue de la Pointe
59139 Noyelles-lès-Seclin, France

[2] RMod team, Inria Lille Nord Europe,
University of Lille, CRIStAL, UMR 9189,
59650 Villeneuve d'Ascq, France

Abstract

More and more companies would like to mine software data with the goal of assessing the health of their software projects. The hope is that some software metrics could be tracked to predict failure risks or confirm good health. If a factor of success was found, projects failures could be anticipated and early actions could be taken by the organisation to help or to monitor closely the project. Allowing to act in a preventive mode rather than a curative one. We were called by a major IT company to fulfil this goal. We conducted a study to check whether software metrics can be related to project failure. The study was both theoretic with a review of literature on the subject and practical with mining of past projects data and interview with project managers. We found that metrics used in practice are not useful to assess project outcome.

1 Introduction

IT companies have a significant number of projects creating a lot of data: project metrics, bugs reports, source code, continuous integration artefacts, production metrics, social network communications, etc. With the emergence of big data methodologies, these companies hope that data science and especially statistics could help them to evaluate their project health *i.e.,* their success or their failure. Healthy projects will speed up the expansion of the company while unhealthy ones can lead to its failure. To achieve this goal, one major IT company asked us to find correlation between metrics of their projects and health of these projects. The hope is that the organization could follow the projects evolution and take preventive actions to avoid project failure. Finding the right metrics in the whole data set is challenging and doing it in a preventive way even more.

Some studies have already been conducted in this field. They usually consider open-source projects and close-sources projects from other companies, but as development environments are likely different, they may not apply our case.

In this paper, we make three contributions. First, we did a literature review on project health predicted by data mining. Second, we experimented more than 10 project metrics on 50 real world, close source, projects. Third, by doing interviews with company project managers, we found indicators that could be linked to project health.

The rest of this paper is organized as follow: in Section 2, we will do the review of the literature. Section 3 presents the result of the data mining of the projects. Section 4 is dedicated to the project managers interviews and conclusion will be presented in Section 5.

2 Literature review

As first contribution, we did a review of the literature. Our hope was that previous work could already have found relationships between project health and project metrics. We found papers that studies both open-source projects and close-sources

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at http://ceur-ws.org

projects. From these findings, we extracted the three most relevant papers.

Capiluppi and Fernandez-Ramil's (2007) goal was to find metrics identifying regressions after refactorings. Easing the maintenance can be considered as improving the health of the project. They used eight open-source software project in C++ to make correlations between four code metrics at function level. They found that no metric alone is a good predictor of regressions. Moreover one has to determine for each individual project which combination of metrics could be used. This study is closer to what we aim to do, they do cross-project correlations. Nevertheless, this study is only using open source software.

Nagappan et al. (2006) described how they apply statistical methods to predict bugs after client delivery. This bug number is not uniform between all the part of the software, so it might reveal some development defects and impact the health of the project. They mine five Microsoft projects, written in C++, and correlate 31 code metrics with post-release failure data retrieved on theses projects. By doing statistical analysis, they found, for each project, a set of metrics related to post-release bugs. This set is changing from one project to the other. It is not possible to apply the same set of metrics to all projects. They found the same results that the previous study, *i.e.,* no set of metrics can be a good predictor alone.

So is it possible in some cases to create a statistical model from one project and have good results by applying it to another project. But it is rare and there is no way to know in advance if it is going to work.

Zimmermann et al. (2009) had for goal to predict class defect from both metrics from source code and project environment. They consider the number of post-release bugs to measure the success of the project. They extracted data from 28 datasets both closed-sources from Microsoft projects and open-source. 40 metrics were gathered. For each metric, they computed median, maximum and standard deviation at class level. Their empirical study gave some results: on 622 cross-predictions between project tested, only 3.5% of the couples can predict each other. For instance, some Open-Source Software (OSS) projects are strong predictors for some closed-source projects but do not predict the other OSS. Some OSS projects cannot be predicted by any of the projects in their study. On the closed-source side, they found some projects that can predict other closed-source projects. However, they also found some projects that do not predict other projects. They also found some systems that are predicting each other.

Others studies are focused not on project metrics, but on social coding.

Wolf et al. (2009) study the link between team communications and the result of the integration process after merging of the developed software parts. It is based on the IBM dataset, Jazz. They studied 1 288 build results, 13 020 change sets and 71 000 comments on 47 different team in 4 months. They used 8 metrics representing the exchanged informations. The authors found no unique measure of social network that indicates if the integration process is a success or failure.

Hu and Wong (2013) examine the influence of social metrics on defect prediction. On six releases of NetBeans and seven of Eclipse, they studied the relations between developers thanks to nine metrics on commits. They studied the impact of these metrics on after release defects. The authors found that the developers relationships metrics are not correlated to the number of after release defects.

Menzies and Zimmermann (2013) reference the progress of the predictive analysis applied on software projects. They precise that it is possible to make studies from various data. However, it is impossible to throw conclusion from a project and apply them to all. As they said: "But more recently, the goal of analytics has changed — the research community has accepted that lessons learned from one project can't always be applied verbatim to another." The research heads towards local methods, *i.e.,* be applied to only one project.

To summarize this section, it seems possible to find, for a given project, metrics that allow one to do predictions *a posteriori*. But finding, *a priori*, a unique metric or a unique combination of metrics that can be applied to all projects seems unlikely.

However, as development environments between companies and open-source are different, we decided to try with the company data and metrics.

3 Data mining

We conducted a statistical analysis on the company projects. Monthly, project leaders fill Excel files containing information on their project about bugs encountered, budget spent, and budget remaining. In these Excel sheets, we have 12 project metrics available for each month related to several category of bug (Critical, major, minor, in qualification, acceptance or production), to the budget, and to the slippage. We used also a metric representing the length of the project name. It is intended as a placebo metric. We will compare all results to this obviously irrelevant metric.

For this company, the project health is related to client delivery. We know that a project succeed if the application is delivered in time, in budget and with the functionalities the client wanted. A project is failing if one of the previous items are missing.

As project slippage metric is the most followed by project leader, we used three metrics related to it *i.e.,* # months in slippage, # days of slippage and if there is slippage or not. We decided to compare these metrics to the metrics related to bug number and budget data.

We used data from 44 projects during the past 3 years. However, only 19 are exploitable because in these, the data are well filled.

By the value of their metrics, several projects can have a great influence on the sample. Statistical methods advise to take out these kind of extreme values to have a better sample to analyse. Two items of the sample were detected as outliers. We moved them aside to conduct the study.

We correlate the 12 metrics two by two in a matrix. The correlation matrix in Figure 1 highlights whether there is a linear dependency between 2 variables.

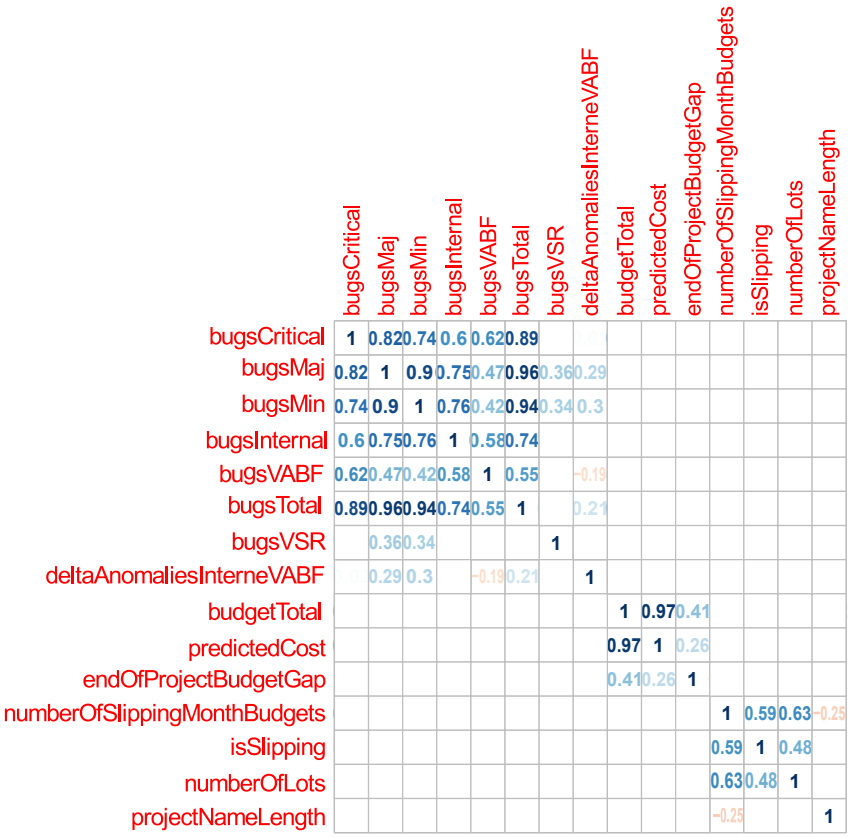

Figure 1: Correlation matrix between each metric of the sample

From this matrix, we inferred three blocks of variables correlated. First, we can infer a strong correlation between all kind of bugs (the darker square at the top left of the matrix). except the number of bugs in production. These bugs are not strongly correlated to the others. It can be due to the fact that it is the final user who found them. The final user didn't take part into the project requirements elicitation step of the project. It is a new eye on the project. So it seems natural that the number of bugs found is not correlated to the other ones.

Second, as shown in the middle square in the matrix, the budget variables are also correlated: the total budget and the predicted budget. The difference between the initial and final budget seems also correlated to the budget metrics but not to the slippage ones. It might be due to the fact that the bigger a project is, the more difficult it is to predict the budget. A long project is more likely subject to deviations.

Third, it seems also the slippage metrics are significantly correlated together which was foretold (except the difference between the initial and final budget). However, the number of intermediate releases seems correlated to these slippage metrics. It might be the decomposition in group of functionalities that is difficult to determine by the project managers.

Finally, there is no link between the 3 groups of variables *i.e.,* the bugs, the budget, and the slippage. Moreover, our placebo metric has worked as predicted. The length of the project name is not correlated to the other metrics.

In the light of this analysis, we can conclude that there is no link between the slippage and any other studied variable. To summarize, the correlations we found are quite trivial. Like the papers from the literature survey, we are not able to find metrics to explain the project health.

4 Interviews

To complete the project study, we realized some interviews with project managers of the company to get their feeling on what impacts project health *i.e.,* success or failure. We wanted to know what are their problems in developing projects, how they detect them and resolve them. The interviews lasted one hour and were decomposed in two parts. In the first, we presented the research topic to the interviewee, in the second, we let totally open the discussion to get all the experience of

project managers on project success and failure.

We met project managers whose projects were used in the analysis and not. We interviewed four projects managers. The projects they lead are diversified. There are both successful and failed projects.

In these interviews, they identified the following root cause of project failure:

- Delay at the beginning of the project: if the client decides to begin the project later, the project team is already available and the relationship will deteriorate.
- Collaboration between the team and the client: if the team and the client know well each other, the collaboration will work fine and the project is more unlikely to fail.
- Team cohesion: if the members of the project team support each other, the cohesion is stronger and the project has significantly more chance to succeed.
- Understanding of the specifications: if the project team understands what the client says and succeeds to transcribe it in is own technical language, the project will progress easier.
- Knowledge of the functional concepts: if the project team knows, in more, the business concepts of the client, the project has more chances of success.
- Change of the framework during the development: if the technical tools or the framework, that the project team uses, change, it will cost more to the project.
- Experience with the used frameworks: a team with experience on the development tools or frameworks they use for their application, will be quite capable of doing the project faster.
- Bypass the qualification tests: if the team doesn't test its application before delivery to the client, by lack of time for example, the client will be unhappy because some functionalities will not work and some tension in the project team will appear.
- High number of bugs listed by the client: as a consequence of the previous item, the client will find more bugs in the application.

These causes of failure are difficult to find in data provided by the project team. That might be why we didn't find any correlation by applying statistics to the projects.

5 Conclusion

A major company asked us to found metrics that predict project success or failure. We conducted a study to check whether software metrics can be related to project failure. The theoretical study of literature shows that the metrics extracted from a project can't be used on another one. The mining of data we have done on company project highlights there is no link between project metrics and data. However, the interviews we conducted shows that the metrics linked to success can't be found by mining project data.

Moreover, as all these studies intervene *a posteriori* on projects, it seems random for a new project to know which metric or set of metrics uses to assess success. Predictive analysis will not work well if it is not possible to know *a priori* which statistical model use.

To go deeper, we plan to do a survey to check whether the indicators we identified during the interviews are shared by all the employees of the IT company.

References

A. Capiluppi and J. Fernandez-Ramil. A model to predict anti-regressive efforts in open source software. In *23rd IEEE International Conference on Software Maintenance*, oct 2007.

Wei Hu and Kenny Wong. Using citation influence to predict software defects. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 419–428, Piscataway, NJ, USA, 2013. IEEE Press.

T. Menzies and T. Zimmermann. Software analytics: So what? *Software, IEEE*, 30(4):31–37, July 2013.

Nachiappan Nagappan, Thomas Ball, and Andreas Zeller. Mining metrics to predict component failures. In *Proceedings of the 28th International Conference on Software Engineering*, ICSE '06, pages 452–461, New York, NY, USA, 2006. ACM.

Timo Wolf, Adrian Schroter, Daniela Damian, and Thanh Nguyen. Predicting build failures using social network analysis on developer communication. In *Proceedings of the 31st International Conference on Software Engineering*, ICSE '09, pages 1–11, Washington, DC, USA, 2009. IEEE Computer Society.

Thomas Zimmermann, Nachiappan Nagappan, Harald Gall, Emanuel Giger, and Brendan Murphy. Cross-project defect prediction: a large scale experiment on data vs. domain vs. process. In *Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pages 91–100. ACM, 2009.