Lessons Learned In Implementing Agile Software Development Metrics

Nilay Oza  
*University of Helsinki*, nilay.oza@helsinki.fi

Mikko Korkala  
*VTT Technical Research Centre of Finland*, mikko.korkala@vtt.fi

Follow this and additional works at: [http://aisel.aisnet.org/ukais2012](http://aisel.aisnet.org/ukais2012)

Recommended Citation

[http://aisel.aisnet.org/ukais2012/38](http://aisel.aisnet.org/ukais2012/38)
LESSONS LEARNED IN IMPLEMENTING AGILE SOFTWARE DEVELOPMENT METRICS

Abstract

The conventional software metrics are not directly adaptable to agile approach due to their intrinsic differences in the focus, goals and process of software development. Subsequently, this paper examines the current state of agile-metrics initiatives in a specific product line of a multinational technology company. Using qualitative interviews based data from five stakeholders the study examines adoption of metrics and as a result provides concrete recommendations toward agile-metrics initiative in the case company.

Keywords: Agile software development, Software metrics, Agile metrics, Large scale software development

1.0 Introduction

Measurement is an essential part of any organization. There is little chance of scalability without directing an initiative that we cannot measure. The scope and objective of any measurement could vary across organizations. In the case software, the metric is a measure of some property of a piece of software or its specifications. It is however difficult to determine what to measure to get the meaningful results? There are loads of pre-defined metrics (Fenton and Neil, 2000) that one can apply but may not lead to any meaningful results or sometime they may add ambiguity and misdirection into subsequent actions of the organization. To add to this scenario, agile methods used in software development (and management) are even subject to a higher degree of mishandling in terms of measuring the results (or progress).

The rest of this paper is structured as follows. Section 2 outlines the background work in agile metrics. Section 3 then presents the research method in brief. Section 4 presents key findings from the case study analysis. Section 5 draws conclusions from the study.

2.0 Related Work

2.1 Why Agile Metrics?
2.1.1 Development Life Cycle
Working with different life cycles is difficult. Most agile methods focus on immediately delivering functionality, which is not feasible in large scale product development of embedded software. Working in an environment, where there are tens of teams involved in developing one product and system contains several products, where there are several customers to the product and where creating functionality which is useful for the customer takes several iterations require some modifications to the agile methods. Every “potentially shippable product increment” is not released to the customer, i.e. in addition to the team level short iterative cycles there is a longer release cycle. Organizations need to define a cycle and rhythm for releasing activities (e.g. creating support and delivery capability).

In fact, agile development makes the entire development cycle much more like a maintenance phase by providing short, focused iterations. Additionally, one of the fundamental characteristics of agile development is an active customer involvement. In small internal IT-projects, it means involvement with an on-site customer. In larger scale product development, it means close cooperation with customers during development and having a customer-proxy (e.g. product manager or product owner – or several of them) working in close cooperation with the development teams. The active involvement of customer brings opportunities for continuous and measureable evidences related to progress. The Product Owner (PO) being responsible for prioritization and being in-charge of the development reduces the need for many traditional metrics for measuring progress of conformance to plan. The PO can go and see the situation without relying on reports and metrics.

2.1.1 Agile Metrics
The metrics used for traditional software development approach are largely unusable for agile approach due to their different view on developing software. The main difference between traditional vs. agile software development is plan driven vs. result driven approach with the focus on iterative process. Designing metrics in agile software development also requires an evolutionary approach – ‘the trick is to iterate. Design a metric. Test it by gaming. Then redesign the metric and test it again’.

2.2 Agile Software Development Metrics
Lessons Learned In Implementing Agile Software Development Metrics

Recent studies indicate an increased interest in agile metrics (Dybå and Abrahamsson, 2008). The classification of metrics has surfaced in different classes in the literature. However, in its collective form, three classes have emerged as: 1) code level (e.g. running tested features, Leffingwell’s iteration and release perspectives, and code quality and design metrics), 2) productivity/effort level (e.g. burn-down charts, and project size units) and 3) economic metrics (e.g. earned business value and break-even point).

The code level metrics can provide visibility into quality of the code, but it is rather uninformative from customer’s point of view and offers almost no help in making decisions about project development. The productivity and economic metrics can support decision-making process but in many cases they too would not reflect the dynamic and iterative nature of the agile development as the sources of information used in measurements (such as work breakdown structure, or earned business value) rest on Product Owner’s estimations or judgments. Figure 1 shows an instance of metrics set relevant to agile. In addition to basic three levels as mentioned above, the figure 1 classifies metrics across seven categories.

Figure 1 shows just a snapshot of agile metrics covering a broad range of metrics that are relevant to agile approach. One of the branches shows strategic metrics. The strategic metrics show examples such as net present value, earned business value (Rausthome, 2006), running tested features and real options analysis. Other categories in Figure 1 include metrics relevant to testing, iteration, automation, code, engineer and project management.
Almost all example metrics in Figure 1 result into quantitative measures based on the piece of software or collected data from the process. Defined quantitative metrics are not always equally applicable throughout all development methodologies, especially when moving from waterfall towards agile development. Please note that metrics mind map shown in Figure 1 is an example, showing the collection of agile metrics being used widely. The company specific agile metrics map may differ from it.

It is always challenging to find quantifiable measures for softer aspects such as motivation, commitment and satisfaction in a certain approach (such as agile) or any other metaphor. Often, the opinion polls or surveys are used to collect perceived knowledge from wider community in the organization and emerging patterns provide basis for representative metrics. Opinion polls fit the notion of regular feedback, also inherent to agile approach. This is certainly a good practice, however, we, based on Klein and Candit (2008) recommend:
Lessons Learned In Implementing Agile Software Development Metrics

- Opinion polls should be conducted regularly to identify challenges and track the success of countermeasures.
- Using a kind of standardized form of a questionnaire, projects or organizations are comparable concerning their problems identified and the speed of improvement or problems to be solved.
- Opinion polls should be automated with a web interface as an add-on or even alternatively to usual interviews.

It is also cautioned to consider complexity and frequency of opinion polls. If executed too often or in a too time-consuming manner, people will easily get tired of answering the questionnaire, which can lead to inaccurate results. Finally, it is important to show and track measures derived from the opinion polls. It positively contributes to people’s motivation when their opinion is taken seriously into account.

2.3 Stakeholder Driven Agile Measurement

According to the principles of stakeholder driven process performance measurement, the best performance is achieved when the goals of all stakeholders are satisfied. This also requires a balanced approach considering viewpoints of different stakeholders. We take Scrum method for this report and present key stakeholder driven metrics from Mahnic and Zabkar’s (2008) work. Indeed, only some of these metrics may be relevant to specific context, however, we propose to note that the stakeholders’ interests might be categorically weaved into metrics collection strategy of the organization.

On the other hand, stakeholder driven metrics may also increase complexity if there are many metrics relevant across the organization. The mixed approach of identifying key measurement interest areas and aligning them with stakeholder driven approach could be highly useful. Prospects of such approach should be discussed and evaluated.
Lessons Learned In Implementing Agile Software Development Metrics

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Goal</th>
<th>Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG1: Product owner and business management</td>
<td>Timely information on project performance</td>
<td>AG1-1: Ratio between work done (e.g., story points) and the planned burn-down.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG1-2: Schedule Performance Index (SPI) (ratio between the earned value [i.e., the value of all tasks completed] and the planned value [i.e., the initial estimate of value of all tasks to be completed till a certain point within the project])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG1-3: Cost Performance Index of Labor Costs (CPI) (ratio between the earned value (measured in units of currency) and actual costs)</td>
</tr>
<tr>
<td>Quality Improvement</td>
<td></td>
<td>AG1-4: Error Density (number of errors per KLOC [Kilo-loc lines of code])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG1-5: Costs of Rework (product of hours spent on reread and cost of an engineering hour)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG1-6: Fulfillment of scope (ratio between the number of tasks completed in the Sprint and total number of tasks in the Sprint Backlog or between the number of PBIs completed in the release and total number of PBIs committed)</td>
</tr>
<tr>
<td>AG2: Team members</td>
<td>Job Satisfaction</td>
<td>AG2-1: The Average Amount of Overtime at Sprint/Release/Project level (the expected hours, the amount of work spent and administrative days)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG2-2: The Average Number of Projects the Employees Work in Parallel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG2-3: Qualitative Evaluation of Working Conditions (communication and teamwork, physical discomfort, psychological well-being, workload, supervision, opportunities for growth etc.)</td>
</tr>
<tr>
<td>AG3: Scrum Master</td>
<td>Efficient Impediments Resolution</td>
<td>AG3-1: Average Number of Impediments per Task/Sprint/Team</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG3-2: Mean Time for Resolving an Impediment (as Task/Sprint/Team level)</td>
</tr>
<tr>
<td>AG4: Customers</td>
<td>Customer Satisfaction</td>
<td>AG4-1: Qualitative Evaluation of Customer Satisfaction (the quality of product, timely delivery, price adequacy, reliability in terms of time and costs, completeness of product delivered at the end of each sprint or release, flexible handling of changes in requirements, good collaboration with the development team, adequate training and documentation etc.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG4-2: Error Density</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG4-5: Estimation Accuracy [Sprint] and Amount of Change [product B])</td>
</tr>
</tbody>
</table>

Figure 2. Stakeholder driven metrics (adopted from Mahnic and Zabkar (2008))

3.0 Research Method

This study is based on qualitative data analysis from the interviews conducted with five stakeholders. The interviews were conducted to better understand the state of agile-metrics at the company and further abstracted in the analysis with emerging patterns. The analysis and data collection went parallel to get maximum value from emerging patterns. The intermediate results were also discussed with the main contact person at the company. The literature review presented in the report went in parallel of the study. The study questions in qualitative discussions came from two rounds of meetings with the relevant company representative and two subject experts. Finally, the findings were structured and organized in this report, which too went through five weekly iterations.
3.1 Data Collection and analysis

We used a qualitative thematic analysis to conduct the empirical study. The experts involved in the study were interviewed using open-ended thematic interviews. The qualitative patterns from the collective interview transcripts were then examined to conduct thematic analysis.

We used standardized open ended interviews to collect qualitative data (Patton, 2011). Standardized open ended interviews are useful when it is only possible to interview participants only once for a limited period of time. Interview questions and overarching objectives were standard for all interviewees. However, interviewees had all the freedom to talk about range of issues within the specific theme question. The spontaneity of interviewees was appreciated within the context and scope of the questions by adopting flexible probing. Open questions also help the researcher to avoid accidentally introducing any of his or her own preconceptions and protect the validity of the data.

3.1.1. Interview process. We conducted interviews at participating companies’ premises. Interviews took place in an adequate environment either in the office of the interviewee or in a business meeting room of the company. The length of interviews varied between 50 and 90 minutes. The researcher introduced themes in the beginning of the interviews but let the interviewee talk quite freely rest of the time. The researcher avoided commenting and accidentally introducing any of her own preconceptions in order to protect the validity of the data.

The interviews were audio recorded and the recordings were transcribed. Due to the extent of the material the narratives were slightly compressed during the transcription; meaningless filler words, repetition and unfinished syllables and utterances were mostly cut out. In addition to the recordings, some written notes were taken during the interviews.

Demographics of interviewees and a participating company are not presented to protect the anonymity of the company information.
4.0 Results

4.1 Key Findings

4.1.1 Collective vision on use of metrics
The study reveals that there is a mismatch between metrics related perceptions among different levels of responsibility. To be more specific, those who are in senior roles and responsible for product and program issues, seem to think that the metrics program and metrics usage (including visibility of metrics) are in good order. Product line management and quality assurance, however, perceived adoption of metrics as not ‘convincing’. It appears that these functions, being closer to the ‘grass-roots’ of actions, have identified quite a few shortcomings on usage as well as visibility into metrics. This also leads to a hypothesis that at the team level (development and engineering teams), the perceived picture of metrics might be very different from the senior or middle level.

4.1.2 Scope of metrics
The metrics seem to be well collected across different functions including quality, product development and program management. One of the key observations in the study shows that the metrics became interests of ‘individuals’ rather than ‘organizational improvement’. This may lead to misguided use of metrics and may also direct actions towards ‘non-value adding’ directions. Some of the other observations related to scope of metrics are:

- Metrics are not viewed to promote awareness of collective progress and integrated into culture.
- At the time of this study, effort is measured with two different data points by different teams, either in ideal hours or in story points. This might impact the scope of metrics.
- The study reveals that there is sufficient coverage of metrics. However, the total velocity (for all teams and whole product) and total technical debt might be useful addition to existing collection of metrics. The correlation between technical debt and team velocity should be examined at all levels (for example, does reducing technical debt increases or slow down the velocity?). Velocity is not comparable between teams, but it could be looked at over time for each team and the whole product. Of course, technical debt is not the only contributing factor to velocity, as
people’s competencies have even bigger role. Therefore this approach is not feasible with very new teams, who still have a lot to learn to master the technology and product.

- Finally, the study recommends Earned Business Value (EBV) metric as it provides needed macro level insight into business value (Hartmann and Dymond, 2006). It appears that, in the current situation, due to unavailability of financial data in the case company, the EBV does not prove its relevance. However, using EBV, an estimated value may still be measured.

4.1.3 Exploitation and impact of metrics

The exploitation and subsequent impact of metrics is quite crucial. This study observes that exploitation of metrics is at micro level contexts and therefore their impacts at organizational level may not be fully utilized. It is clear that the adoption of agile approach has increased visibility in the development. However, the impacts of agile adoption are mostly immeasurable and based on the respondents personal perceptions. The attempts of collecting and integrating perceived or tacit knowledge should be more regular (rather than sporadic) as it will provide an opportunity to see the trend and subsequent directions for metrics exploitation.

- In several instances, work is mostly centred on solving daily and unforeseen priorities from different stakeholders. Something that can be trusted and fosters visibility is lacking. This is not to argue against agile attitude of being reactive to changes but constant and too many directional changes bring instability.

- There is an overload of work in some instances. The continual overload may result in less productivity and ‘de-motivated’ attention to measurement culture.

- Results of different metrics are just used as they are. It should be noted that developing an understanding of reasons for variability in the data may provide more insights than mere quantitative results. For example, looking at graph showing variation between size and effort will tell much more if there is a developed understanding on what’s behind the variation!

- There seems conflict in utilization of metrics. The conflicts, whether at group level or team level, may also act against the collective vision.

- In one instance, it seems that teams do not have mature view or thorough understanding on what agile development means. Some teams have misinterpreted agile development as meaning that creating new functionality is the only
important thing, and that quality is not important. This has resulted into discarding practices like code peer-reviews and not fixing bugs as they emerge, which have ultimately resulted in decreased quality. The coaching support required by teams and management should be identified and arranged.

- Coaching on better estimation may be helpful to develop more meaningful insights from metrics. Variances in estimation may be reduced by coaching and as teams get more experience. At the time of this study many teams were relatively new to the product.

4.2 Key Recommendations

In this section, we present six concrete recommendations that may help in improving measurement in agile software development.

1. The impact of metrics should be made visible at all levels.

2. The collective vision on agile-metrics should be intensified on breeding the culture of measurement.

3. There may not be a need for collecting several metrics. The company should understand the reasoning, relevance and importance of collected metrics.

4. There should be a dashboard highlighting all major indicators of the development. If you have the dashboard, then, it should be visible and used throughout the organization. An executive dashboard reflecting only most important indicators may also be developed.

5. Teams want visibility into what’s happening holistically. Additionally, too many unforeseen tasks may reduce the total velocity of teams and increase technical debt. This should not be left un-examined.

6. Rather than following too many metrics, only one metric should be tuned in for a specific period of time. This may also help building measurement culture. It has also been noted in other successful metric implementations that following one metric for a specific period of time builds capability in the team for that specific metric. The metric then gets properly ’attuned’ rather than becoming the victim of ’compliance’!

4.3 Limitations of the Study

This study has at least two limitations. The study indeed suffers from purely qualitative data based analysis. The study is also limited in the sense that the key
findings and recommendations are for the specific case. They cannot be generalized for generic metrics implementation scenarios, even if the company is using an agile approach. We tried to increase validity of the study by analyzing qualitative data from multiple sources. We also feel that the case under study being a real agile metrics implementation environment, offered more hands-on perspectives and qualitative data, which is rather difficult to avail.

### 5.0 Conclusions

This study investigated agile-metrics initiatives in a specific product line of a multinational technology company. The study used qualitative discussions and identified key observations related to the use of metrics and overall vision behind the metrics related initiatives. It is clear that there is no shortage of metrics being collected to measure progress of the product development. We also focused on information sources, on which key stakeholders make decisions, how metrics are selected and impact of agile adoption in general.

The results of this study indicate that there is a need of inducing a reference framework for metrics program. It is not sufficient to merely collect all possible metrics but driving the culture of continuous measurement is imperative. This may also help to build collective vision on metrics, which seem to have some contingencies across different levels of responsibility. The study has made specific recommendations in the report.

The study also reveals that agile adoption have brought significant positive impacts to product development and coordination. The efforts on strengthening agile adoption capabilities should be continued. However, the current impacts of agile adoption perceived by different stakeholders are lacking the solid evidences to support the respective claims. The agile adoption related surveys which aggregates qualitative perceptions on agile should be continued at regular intervals. Measuring impacts on soft aspects such as motivation, commitment and trust are in any case challenging for quantification. The aggregated qualitative patterns over longer term will certainly increase its credibility.
Lessons Learned In Implementing Agile Software Development Metrics

References


Patton, M. Q. (October 2001). *Qualitative research and evaluation methods*, Sage Publications (3rd ed.).