Seven Important Metrics for Agile Performance
By Mike McEwen, PMP, Dynanet Director of Software Solutions

Agile is an empirical approach to performing work. Instead of following a detailed plan developed upfront, the Agile team observes and experiments, a process that metrics directly support. Metrics are something useful to measure, like customer satisfaction or network bandwidth. I use metric interchangeably with measure and call the actual values, measurements. Examples of measurements include a rating of 5 on a scale of 1-5 for a customer satisfaction metric or 100 megabits per second for a network bandwidth metric. A Key Performance Indicator (KPI) is a metric or measure that reveals something about organizational performance (i.e., how well a group, office, or team is doing) and often has a related goal, target, or threshold (e.g., a rating of 4 on a scale of 1-5 for customer satisfaction and 80 megabits per second for network bandwidth). Measurements can be point values but become more useful when viewed over time.

In the Agile world, metrics help you establish baselines for comparison when making changes; monitor team performance; determine if things are getting better, worse, or staying the same; communicate status or issues transparently to project stakeholders; and confirm assumptions and the effectiveness of “experiments” (i.e., improvements tried). One typical Agile metric is velocity, the number of story points a team completes in a sprint or iteration. It represents the rate or throughput of work the team delivered in the past. Measurements of velocity can vary widely from one iteration to another because of changes to a team (absences, people joining, etc.) and the type of stories the team handles (new features versus bugs), so look at averages of this metric over time and be careful about drawing conclusions. Don’t forget that, in general, you cannot compare the velocities of separate teams because each team has a different understanding of what a point means.

What are some good Agile metrics? Pick KPIs that are easy to calculate or, better yet, calculated by a sprint management tool automatically. Ideally, the tool will also display the metric values graphically over time to show trends. I recommend the following seven Agile metrics:

1. **Sprint and Release Burndown/Burnup**. Like velocity, sprint and release burn rates are standard metrics you should track, share, and discuss with the team. They report the work completed over time (it fluctuates) and work remaining. Resist the urge to measure work in hours for this metric. Instead, use story points to show the team’s progress toward their goal, which is to deliver working software not spend hours. Sprint management tools commonly overlay the team’s progress with lines representing the ideal burndown, required velocity, and some quantification of the current backlog (e.g., story points remaining).
2. **Size of Backlog**. Expressed in story points and viewed over time, this metric helps everyone understand what is happening to the queue of future work. While the team may want to see a product backlog increase over time for job security, Product Owners and end users may become frustrated if features are not delivered in a reasonable timeframe. The release backlog should decrease as the team finishes iterations and can help predict when the release will be done.
3. **Lead and Cycle Times.** Lead time measures how long the team takes to deliver a working feature after a related item is added to the product backlog. Because some low-priority items may stay on the backlog a long time, consider excluding low priority items from the calculation or weighting the average by some quantification of priority. Cycle time is a related metric that measures the length of time it takes a team to deliver a feature after starting to work on it. Comparing cycle time with the initial estimates of related development tasks can reveal insights into your estimation process.

4. **Number of Defects Found.** I track the number of defects found during testing and in production. Values of these metrics reflect the quality of code being produced as well as the effectiveness of testers, automated test scripts, code reviews, and other quality techniques. This metric can also help you maintain an effective developer-tester ratio.
5. **Team Reliability.** Expressed as a percentage, team reliability is the average number of stories or points attempted divided by the number actually delivered. This metric indicates the accuracy of the team's estimates and the confidence stakeholders have in the team meeting its commitments.
6. **Team Satisfaction.** Conduct an anonymous poll after every sprint asking team members to rate their happiness subjectively. For example, "On a scale of 1 (lowest) to 10 (highest), please rate your satisfaction with the team's performance." Average these ratings across the entire team, plot them over time, and discuss during retrospectives.
7. **Customer Satisfaction.** Ultimately, the satisfaction of end users, executive sponsors, and other customers will determine the fate of the team. Ask stakeholders to rate the team using "net promoter score" or a similar measure on a regular basis to make sure the team is meeting expectations and delivering value to the customer base or organization.

Besides these metrics, you might also find traditional project management measures related to cost and time useful, such as budgeted versus actual cost and percent completion of tasks. Whatever you use, remember these points:

- Don't use metrics that evaluate individuals or demoralize the team.
- Always wait until a team is established before drawing any conclusions.
- Trends are more important than specific values.
- Metrics are not an end; they are the beginning of a conversation with the team and other stakeholders.
- Metrics should ultimately improve the quality of the team's deliverables and happiness of customers.
- Take an Agile approach to metrics: try some, evaluate their effectiveness, and change if needed or as the team matures.

Tracking metrics and following this guidance will enable better transparency, inspection, and adaptation—all traits of successful Agile projects.

Dynanet Corporation, a minority-owned small business located in Elkridge, Maryland, has provided a range of information technology (IT) services to multiple Government agencies. Our focus on customer satisfaction and performance has established long-term relationships with the United States Air Force (USAF), Food & Drug Administration (FDA), the General Services Administration (GSA), the Office of Personnel Management (OPM), and the States of Maryland and Pennsylvania. During the last few decades, our executive leadership team has built Dynanet's dedicated workforce from 20 employees to over 125 key personnel. Dynanet is committed to employee engagement and maintains a retention rate of over 90%. Our managers are PMP certified and our employees maintain many sought after professional certifications, such as Certified Scrum Masters and ITIL certified. Additionally, Dynanet has been appraised at CMMI Maturity Level 3 (ML3) and our quality management processes have been audited and found to be in compliance with the ISO 9001:2008 quality standards. Further, our company has a Top-Secret facility clearance. For more information, please visit Dynanet's website at www.dynanetcorp.com.

About the Author: Mike McEwen (PMP, CSM) is a servant-leader who forms amazing teams and helps them deliver award-winning technology solutions to commercial, non-profit, and government customers.