DYNANET CORPORATION

Eight Tips for More Rewarding Retrospectives
By Mike McEwen, PMP, Dynanet Director of Software Solutions

One of the *Principles Behind the Agile Manifesto* is "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly." This reflection is often called a retrospective and ideally occurs on the last day of a sprint or iteration after a sprint/iteration review but before the next one is planned. As a feedback loop, retrospectives directly support the "inspect-and-adapt" aspect of Agile by giving the team the opportunity to review its own performance and identify any improvements needed. But retrospectives are not only about fixing problems. Healthy retrospectives also identify best practices that can benefit other teams, reinforce the team's sense of ownership, and cultivate its self-organizing nature.

The agenda for a retrospective is simple. The facilitator asks what went well (positives), what did not go well (negatives), and what should change going forward (deltas). Where the sprint review focuses on **what** the team built, retrospectives explore **how** it was built. Once the team gets into the retrospective habit, though, they may get too routine or boring. The same problems may surface again and again, the team may run out of things to say, suggested improvements might never be implemented, or the meeting may devolve into an opportunity to vent about problems outside the team's control.

Here are some ways to make your retrospectives more rewarding:

1. **Mix it up.** Vary the format, location, and facilitator to keep retrospectives interesting. If you have remote workers in other time zones, vary the time too. Take time to celebrate a sprint or release if it was successful and give public kudos for exceptionally good performance.
2. **Involve everyone** especially if some team members work remotely. Reach out to people who don't contribute to the discussion and make sure everyone on the phone can hear. Use webcams and screen sharing tools. Don't let one person dominate the meeting; ensure that all participants have equal time to contribute. Finally, don't focus on just one issue or only the negatives.
3. **Capture, assign, and track action items** to completion. Every action item should have an owner. Display the list in a public place and put stories or tasks on the backlog to implement them. It's okay if the discussion "free wheels" but the facilitator should help the team draw conclusions and identify specific improvements after enough discussion has occurred.
4. **Ask questions** to jumpstart suggestions. Areas to explore include technical approaches, interactions with external teams, the performance of each team role (e.g., developers, testers, etc.), technical debt, velocity and other performance metrics, and accuracy of estimation. Sample questions the facilitator might ask are "How did everyone like the new automated testing tool?", "Did the developers deliver the code in time for the testers?", "Was the Product Owner available when needed?", "Why was our velocity so low?", "Why did we deliver a lot more story points than what we committed to in the sprint planning session?", and "What did everyone think of the feedback we got during the sprint review?".
5. **Don't hesitate to call out challenges** of specific team members, but keep criticism constructive, never publicly shame, and never forget that the team is ultimately responsible for meeting its commitments despite what individuals choose to do. Recognize that it may be hard for some people to admit their own mistakes, receive personal feedback, or identify the issues of others. Promote an honest and transparent environment where everyone feels respected, supported, and protected.
6. **Discuss retrospectives during your retrospectives!** For example, send out anonymous surveys before a retrospective to see what people think of them, discuss the results during the

retrospective, and suggest improvements if needed. Or, conduct a live poll during the retrospective using a tool like Poll Everywhere.

7. **Consider facilitation tools** like Boardthing, Retromat, Retrium, or Conteneo Weave.
8. **Consult resources** like retrospectivewiki.org for additional ideas.

Some final advice -- never go more than a month between retrospectives and seriously consider having a separate retrospective at the end of a release or project to capture overall lessons learned. Adopting some or all of these tips will keep your retrospectives interesting and, more importantly, allow the team to improve its performance.

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.