Honeywell

The Exponent

Volume 8, Issue 2

Glad-Sad-Mad: A Simple Method to Improve Team Performance

Ed Ambrose
Technical Manager
Honeywell Aerospace
ed.ambrose@honeywell.com

Abstract—**As a manager I've experienced the difficulties of improving individual performance. Improving individual performance can be a difficult task, however, improving team performance can happen much faster. Agile methods and the Scrum framework use various methods to improve team performance. The Scrum retrospective meeting and a method called "Glad-Sad-Mad" can quickly identify factors limiting team performance and determine strategies to remove these impediments. This article provides a basic overview of the Scrum framework to give an understanding of how retrospective meetings fit into the iteration agenda. It also discusses how to use Glad-Sad-Mad and some of the psychology around its effectiveness**

I. INTRODUCTION

I recently finished reading "Misbehaving: The Making of Behavioral Economics" by Richard Thaler and I was struck by the similarities between the transformation in economic theory and advancements in my own field of computer science and software development. For many years a mathematical approach dominated economic theory. However, Thaler kept finding examples where humans didn't behave as the mathematical models predicted. Over time, he and his collaborators (most notably Daniel Kahneman and Amos Tversky) documented many biases that influence human decision making. Initially, their ideas were not well received by the academic establishment and in some cases actively sabotaged. After many years and through the contributions of many additional researchers, their theories have survived and are applied in many useful ways.

Prior to joining Honeywell in 2008, my personal experience using traditional software development methods was also filled with anomalies. Initially, I attributed these anomalies to my inexperience with the tools (Microsoft Project and waterfall project management). I remember measuring things like "How many requirements changed during the design phase?" We called these "requirement leaks"….things that somehow leaked out of the requirement phase that should have been prevented! Often they became excuses for blown budgets or schedules.

As I became more experienced I began to appreciate just how often our design decisions were wrong. For me, this was very troubling. I was surrounded by some of the most talented developers I'd ever worked with and yet many of the products contained serious flaws the impacted their commercial success. Then I read this quote …

> "A *changing requirement is evidence of learning.*"Unknown

This single sentence captured everything we were doing wrong in our approach to software development. For me, it was the beginning of some deep soul searching and a quest to find better methods. Fortunately for me, there were many others who were also unsatisfied and were willing to try something different. Similar to Thaler's experience, the Agile forefathers realized the importance of human behavior and human interactions[1]. There are various ways the Scrum framework and other agile methods improve human interactions. My favorite, called Glad-Sad-Mad, exemplifies how team performance improves when we take into account the human element. Before explaining Glad-Sad-Mad, a brief introduction to the Scrum framework is in order. If you are familiar with Scrum, skip to section III.

A. Scrum Basics

Scrum is a framework for agile development. The word *framework* describes a set of best practices rather than a rigid process. This flexibility allows teams of varying maturity to use practices that best meet their needs. The following illustrates the basic framework:

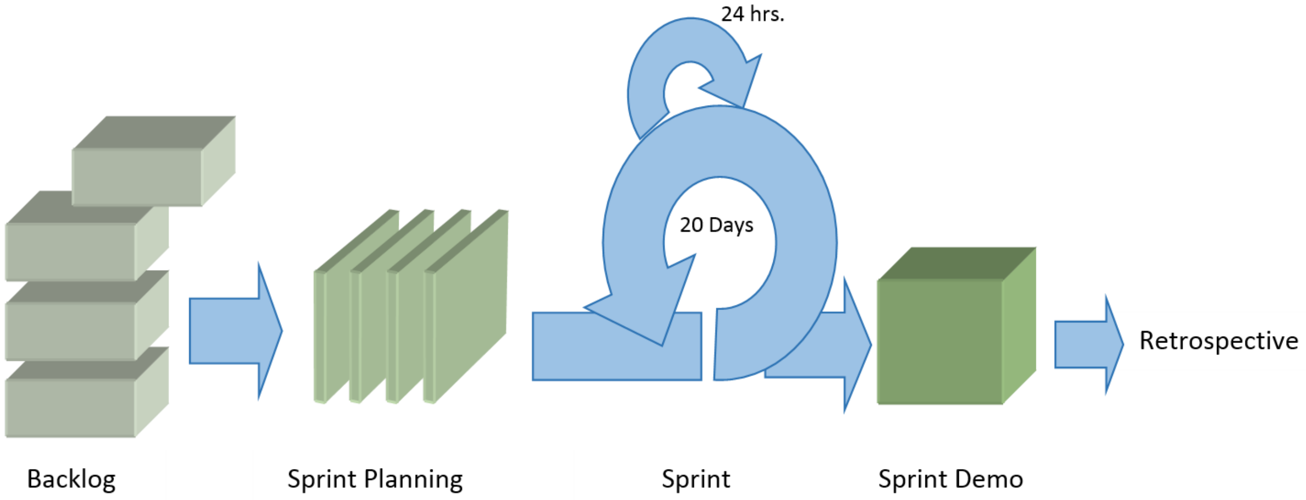

Fig. 1

Scrum uses a repeating fixed duration work cycle and each of these cycles is called a *sprint*. For example, Minneapolis Aerospace scrum team uses a 20 working day sprint. At the end of 20 days, the sprint ends even if there's unfinished work and a new 20-day sprint begins. Sprints duration can vary from 1 week to no more than 30 days. Each sprint is comprised of a Sprint Planning Meeting, Daily Standups, a Sprint Demo, and a Retrospective meeting.

Scrum defines a role called the "Product Owner" (PO). The PO is responsible for the value delivered by the team. For a software team, the value is measured by a "working increment of software that can be demonstrated". At the start of a sprint, the Product Owner must communicate to the development team what is most important (i.e. needed features, bug fixes, etc.) based on the *best available information* at that time. To facilitate this communication, the PO uses a prioritized list called a "Product Backlog". The PO adds items to the backlog and sets each item's priority using input from external stakeholders and the development team. Backlog items come in many forms. They can be descriptions of product capabilities (sometimes referred to as "stories") or they can be non-functional capabilities such code refactoring.

A Sprint Planning meeting starts the sprint cycle and involves the Product Owner, Development Team and sometimes stakeholders. The Product Owner explains each item in the backlog and the Development Team decides how many items they can complete in the next sprint. The team will also define goals for the sprint.

With the sprint plan and goals in hand, the development team starts the next sprint. They self-organize to determine who will implement each item in the plan and they meet daily to discuss their progress and any impediments preventing progress. The daily meetings helps the team stay informed and develop a plan for the next 24 hours. Key to any agile framework is the notion of *inspection* and *adaptation*. The daily standup meeting agenda supports inspection of the last 24 hours and then uses that information to adapt the plan for the next 24 hours. Daily standup meetings benefit from a good visual management system. We use a simple magnetic whiteboard with 3 columns…PLAN, DOING, DONE. Each item in the sprint plan is represented by a slip of paper attached to the board with a magnet. If an item is "blocked", it's identifed with a red magnet or red tag.

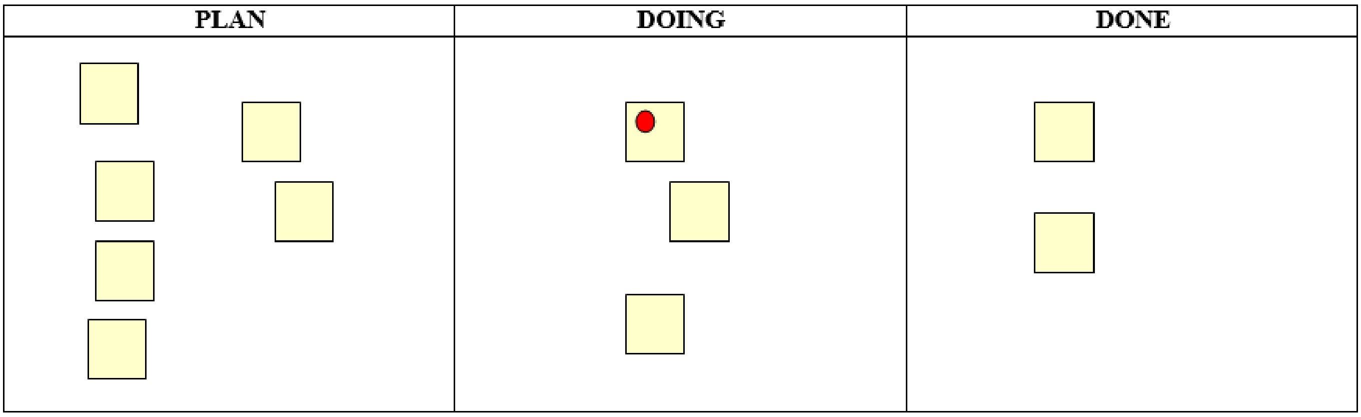

Fig. 2

Scrum defines a second role called the Scrum Master (SM) also referred to as a Scrum Coach. The SM is responsible for the performance of the team. If a team member(s) can't make progress due to an impediment, the SM is there to help. If the team is being interrupted by outside forces, the SM can help protect the team. The SM is in a good position to defend the team from interruptions because everyone (stakeholders included) recently agreed to to the sprint plan. The Scrum Master is also there to observe the team (more on this later). As a general rule, the manager should NOT fill the Scrum Master role.

Once the sprint completes, the Product Owner, development team, and stakeholders meet for a demonstration to observe the working software. This is another *inspection* and *adaptation* point in the Scrum process. The stakeholders inspect the working increment and provide the Product Owner and development team feedback. The PO and development team adapt by adding or modifying items in the product backlog.

The Scrum demo meeting is one way Scrum embraces (welcomes) changing requirements. The inspection process brings to light new information that lets the team create a better plan for the next sprint. As Doug Shimp likes to say….

> *"What ever plan you have….it's wrong! Scrum helps you find right faster."*

> *"Scrum does not prevent problems…it helps you find problems faster while you still have time to adapt."*

Jeff Sutherland's words are a little more harsh…

> *"Change or Die."* [2]

The product backlog is the primary instrument to inject change into the plan and feed the team ~ 2 sprints worth of sprint ready items. Items move in, up, down, or out of the product backlog based on "the best available information" gleaned from the various inspection/adaptation points in the framework.

The Sprint Retrospective is the final meeting in a sprint cycle and represents another *inspection* and *adaptation* point. It's a chance for the team to step away from the daily grind of getting things done and reflect on "how" things are getting done. In my experience, the Sprint Retrospective can have the most impact on long-term team performance. The Scrum Coach usually moderates the retrospective meetings and there are several methods for making these meetings effective. Section III details an excellent retrospective tool called Glad-Sad-Mad.

The above introduction touches on the basics of the Scrum framework. See the reference section for additional recommend readings.[1][2][5][7][8][9]

II. GLAD-SAD-MAD (GSM) MECHANICS

This section shows the detailed steps for conducting an effective retrospective meeting using the Glad-Sad-Mad (GSM) method. GSM requires some pre- and post-meeting work by the *team coach* along with 4 key steps involving the entire team:

1.	Pre-Meeting Prep. (*coach only*)	(5 min.)
2.	Write observations on Sticky notes	(15 min.)
3.	Post & Group observations	(7-10 min.)
4.	Voting	(5 min.)
5.	Discuss Solutions/Next Steps	(15 min.)
6.	Post-Meeting Activity (*coach only*)	(30 min.)
7.	Implement Solutions	(ideally completed during next sprint)

Step 1: Pre-Meeting Prep (coach only)

Prior to the completion of the current sprint, the coach schedules a time and place for the retrospective meeting. If the team uses a Scrum/Kanban board, it's best to conduct the meeting in the same area. It's also a good idea to train any new team members prior to the meeting.

Before you begin, make sure you have these items on hand:

1. Blank White Board with sufficient space for the entire team to gather comfortably*. Label 3 columns on the board with the words "Glad", "Sad", and "Mad".
2. Sufficient quantity of 3" x 5" Sticky notes
3. Pen/Pencils…one for each participant
4. Voting tools (some teams use colored stickers. we use colored magnets….3-5 per team member)

* Open space allows the team members to move about and interact with the board during the meeting. Cramped spaces can inhibit team participation.

Step 2: Write observations

Assemble the team in front of a white board and pass out pens and Sticky notes. For ~10-15 minutes, the team members silently write observations on Sticky notes. Team members should limit the scope of their observations to things that happened during the last sprint. The observations should fall into one of three categories. *Glad* items are behaviors, tools, processes or methods that were beneficial and the team should continue doing in the next sprint. *Sad* items are activities that were within the team's control but the results were not acceptable. *Mad* items are activities that were outside the team's direct control and results did not meet expectations.

Step 3: Post and Group observations

For the next ~10 minutes, in Round Robin fashion, each team member describes a single observation and places it on the GSM board in the appropriate column. As each team member takes their turn, they can group their item by posting next to other similar items. At the end of the 10 minutes, the coach can draw circles around grouped observations and title the group.

Step 4: Vote

For the next ~5 minutes, the team votes on where they would like to focus their improvement activity. Each team member receives 3-5 colored magnets. We use blue magnets but any type of visual indicator would work (i.e. stickers, etc.). Votes are cast by placing a magnet on a Sad or Mad observation.

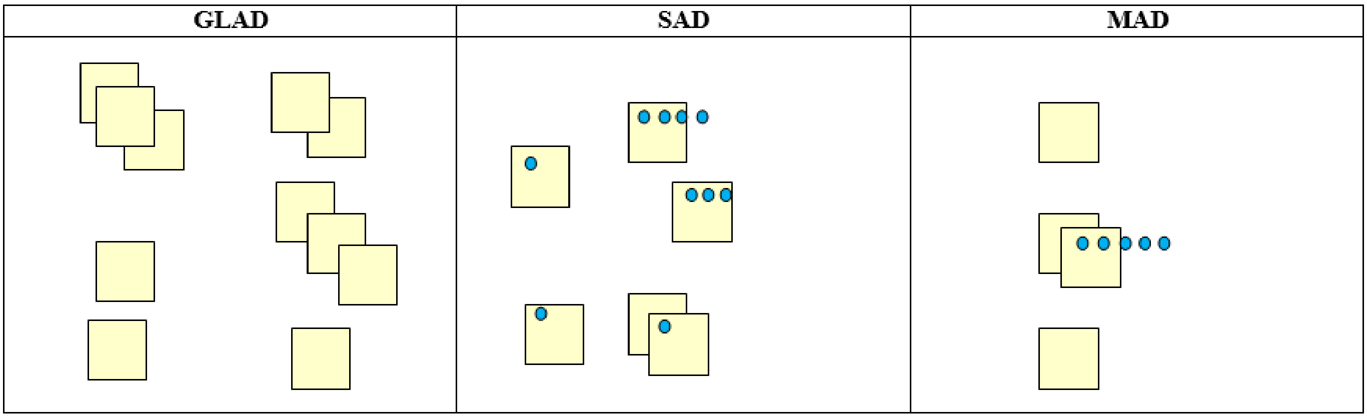

Fig. 3

Step 5: Discuss Solutions/Next Steps

In the remaining time, the team discusses the items with the most votes and how they might improve (i.e. adapt) in the next sprint. Often the team can immediately address the "Sad" items without assistance. The "Mad" items are much more challenging and often represent *organizational impediments*. The coach is responsible for making suppliers and stakeholders outside the team aware of these impediments and working toward common solutions.

Step 7: Post-Meeting Activity (*Coach and Product Owner only*)

After the GSM ,exercise the Coach works with the Product Owner to add improvement ideas to the Product Backlog and properly prioritize. As a coach, I recommend teams implement at least one improvement idea each sprint. This helps develop a *habit of continuous improvement* and sends a clear message that improving team performance is important.

III. The Psychology behind Why It Works

It's important to understand why GSM is effective and some of the psychological reasons that make it helpful. The first and obvious reason is the *simplicity* of GSM. It only takes 5 minutes to train a Scrum team or new team member and within minutes they can be using GSM. Its simplicity exploits "Dude's Law."[10]

$$Value = \frac{Why}{How}$$

Dude's Law

The simple mechanics of GSM minimizes the denominator (the How) and lets the team focus on activities that directly increase the numerator resulting in a more valuable activity. Which would you rather have? A team that spends 45 minutes talking about a tool/process? Or talking for 45 minutes about what the tool/process is revealing and what to do about it? *Simplicity* is a recurring theme in Scrum. For example, the combined length of the Scrum Guide and Nexus Guide (how to use scrum with multiple teams) is 27 pages.

The second reason GSM is effective is the ability to prevent *social loafing*. Social loafing is the phenomenon of a person exerting less effort to achieve a goal when working in a group than when working alone. [3][4] It's awkward for a team member to not participate during steps 2 and 3 of GSM and I've never observed social loafing during a GSM exercise.

Concurrency is the third reason GSM is effective. During step 2 all team members are generating observations. The large number of observations generated in such a short time period always impresses me. On average, the 7 person Minneapolis Scrum team generates ~33 observations in step 2. Not all of these observations are unique but that leads to the next benefit.

GSM helps create team *alignment* by grouping similar observations (Step 3). If two or more people make the same independent observation, they are more likely to be in agreement on where to focus. The voting activity in step 4 provides an additional step in gaining team alignment.

The final benefit of GSM is its ability to avoid *priming* and *regression-to-the-mean*. If a retrospective meeting chooses forego GSM, stronger personalities within the team can have a negative effect. Let's assume we have a normal distribution of possible observations on a spectrum from "not interesting" to "very interesting". The probability of "mildly interesting" observation is higher than the two extremes. The first verbalized observation has a good chance of being only mildly interesting and this "primes" the others to think along similar lines causing subsequent ideas to also be grouped around the mean. GSM helps identify the very interesting observations because step 2 keeps other observations hidden and team members are allowed to think independently. Scrum teams often uses a similar approach to estimating the size of stories using an activity called "Scrum Poker" that keeps the estimates hidden until all the team members have silently made their estimates.

IV. Conclusion

GSM is one of many activities that improves human interaction and can significantly increase the speed at which teams deliver value to their customers. When combined with a continuous improvement cycle such as Scrum sprints it creates a sense of *forward momentum*. Often, the Glad observations are public expressions of thanks to fellow team members. If implemented properly, the "Glad" column also contains solutions to issues that once occupied the other two columns. This can be a source of *intrinsic motivation* driven by three factors…*Autonomy*, *Mastery*, and *Purpose* [5]. Teams that use GSM have *autonomy* to decide how to address issues that impact their performance. They are able to observe the results of these actions (often experiments) and determine if their actions were effective…this experience leads to *mastery* of their craft. Finally, GSM helps the team define a common *purpose* and alignment around addressing impediments.

Final *words of caution*. Like any activity, GSM and Retrospectives can be abused, or worse yet, ignored. Managers and leaders are responsible for addressing *organizational impediments* identified in the MAD column. Failure to address these issues can demotivate the team. Another challenge is virtual teams. I'm often asked, "How does this work with virtual teams?" Answer: "Not as well." However, I've used it with remote teams and it was helpful.

Hopefully, this introduction motivates you to investigate and try agile methods to experience for yourself the joy of self-organizing teams that delight their customers. It requires a healthy dose of humility because whatever plan you have…it's wrong. Doug Shimp likes to say…

> *"Reality Trumps Expectations, so when reality and expectations don't match, it is the expectations that must change."*[8]

Those are hard words to swallow for managers like myself who are accustomed to setting expectations. The agile framework allows stakeholders to set different expectations and still hold teams accountable. The expectation changes to 'How do we deliver *the most value* within our schedule and funding constraints knowing that requirements will change as we learn?' Jeff Bezos said it best when addressing his shareholders:

> *"I believe we {Amazon} are the best place in the world to fail (we have plenty of practice!), and **failure and invention are inseparable twins**. Most large organizations embrace the idea of invention, but are not willing to suffer the string of failed experiments necessary to get there."*[6]

I view Dave Cote's words "Right and Fast" as a *result* of our efforts, not a method. Finding "Right" quickly requires experimentation (failure and invention).

GSM is one way for teams to safely observe failures and quickly adapt, inspect, and adapt."Change or die."[2]

ACKNOWLEDGEMENTS

The Minneapolis ETS/PSE Scrum team for making me a better coach: Marlene Hull, Mary Knickelbine, Val Landwehr, Jonathan Kosbab, Dean Samuelson, Chris Gapinski and Travis Hieb.

Tom Johnson (Staff Software Engineer, Clearwater) for his willingness to try GSM with his remote team members and for his thoughtful feedback on this publication.

REFERENCES

[1] Agile Manifesto http://agilemanifesto.org 2001.
[2] Southerland, Jeff, Southerland, J.J.; Scrum The Art of Doing Twice the Work in Half the Time. P. 10,36,37,
[3] Karau, Steven J.; Williams, Kipling D. (1993). "Social loafing: A meta-analytic review and theoretical integration". Journal of Personality and Social Psychology 65 (4): 681–706. doi:10.1037/0022-3514.65.4.681. ISSN 0022-3514. "the reduction in motivation and effort when individuals work collectively compared with when they work individually or coactively"
[4] Gilovich, Thomas; Keltner, Dacher; Nisbett, Richard E. (2006). Social psychology. W.W. Norton. p. 60. ISBN 978-0-393-97875-9. "The tendency to exert less effort when working on a group task in which individual contributions cannot be measured"
[5] Pink, Daniel H.; Drive: The Surprising Truth About What Motivates Us. 2009. ISBN 978-1-59448-884-9
[6] Amazon.com Inc. SEC filing FORM 8-K, April, 5 2016. Exhibit Number 99.1 2015 Letter to Shareholders.
[7] The *Scrum Guide* available from www.scrum.org
[8] Exploring Scrum: The Fundamentals, Dan Rawsthorne and Doug Shimp. p. 4.
[9] Artful Making: What Managers Need to Know about how Artists Work, Lee Devin and Robert D Austin
[10] Hussman, David. Dude's Law. http://devjam.com/2010/08/05/dudes-law-gordon-pask-shoveler/

ABOUT THE AUTHOR

Ed Ambrose is a 35 year veteran of the technology industry having worked in various technical and leadership positions at Fortune 500 companies and 4 startups. He received his Bachelors of Science (Computer Science) degree from the University of Minnesota in 1981. Ed joined Honeywell Aerospace in 2008 and leads a team of 17 engineers and technicians with Electrical, Software, and Metrology expertise. His team develops and supports custom solutions that fabricate/test guidance and navigation products such as Ring Laser Gyros, Micro-electro-mechanical Systems (MEMs), and Radar Altimeters. Key among Ed's many skills is his ability to lead teams through business transformation and process improvement with the pragmatic use of Six Sigma, Lean, and Agile methods. Ed received his Scrum Master Certification in 2014 and has been using agile methods since 1998. In his spare time, Ed enjoys photography. His mono-prints have appeared at exhibitions in New York, Chicago, and St. Paul, Minnesota.

Honeywell