

The Simple Power of *Feedback Loops*

Colleen Johnson



- CEO & Founder of ScatterSpoke, a team space for more effective retrospectives
- LeanKanban Accredited Kanban Trainer & Kanban Coaching Professional
- Former AgileDenver Board of Directors
- Co-chair 2016 & 2017 Mile High Agile Conference
- Member Agile Uprising Board of Directors
- Mama to three amazing kiddos



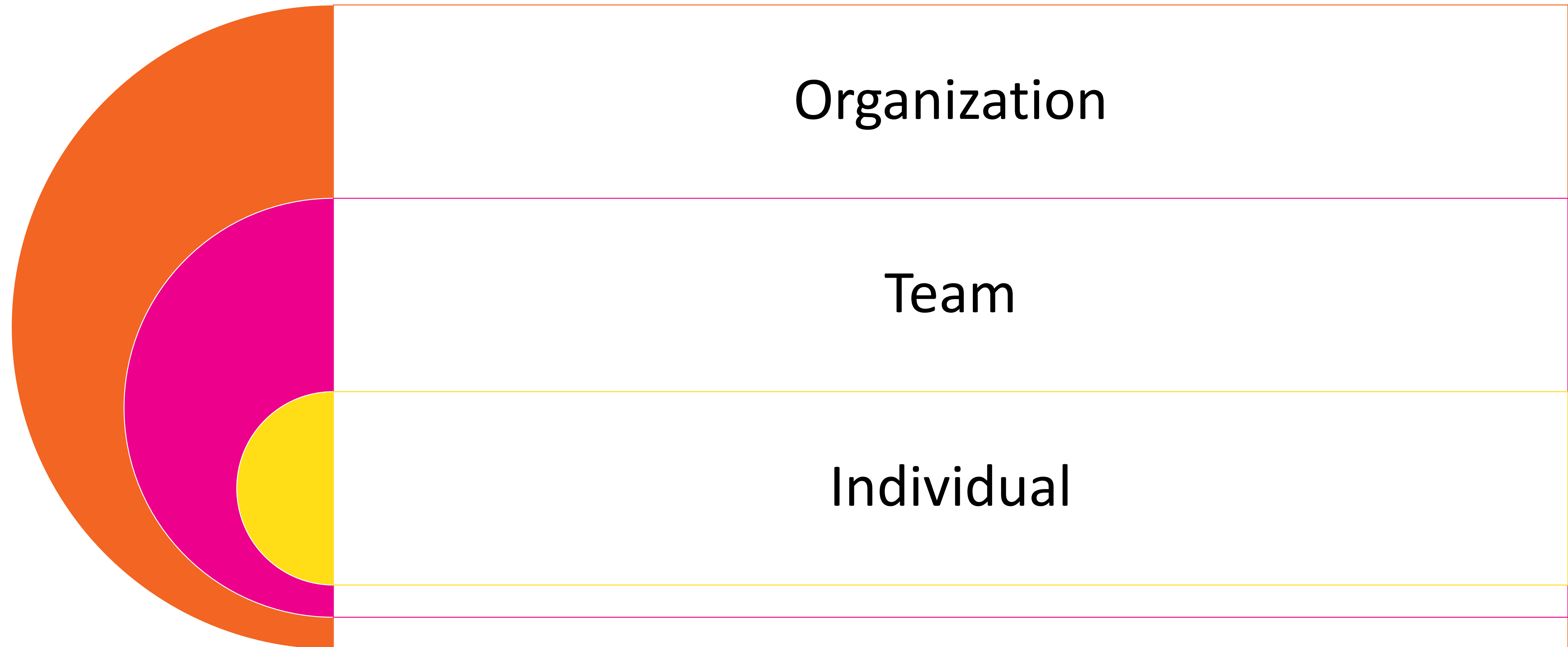


What is a Feedback Loop?

using information about reactions to a product, process, performance, etc. which is used as a basis for improvement.



Levels of Feedback





Individual Feedback

information given to a person about prior behavior so that the person **can make adjustments** in the future.

Giving Feedback



- Dig into your motivations
- Be timely and specific
- Create space, avoid the drive by
- Build trust!

Receiving Feedback



- Understand where the other person is starting from
- Ask questions and dig in to the details
- Assume good intentions
- Don't react too quickly
- Build trust!



Team Feedback

Information about a groups behavior that seeks to drive changes and improve communication.



Why we Retro

- Inspect on current practices, process and team dynamics
- Build open channels of communication and trust
- Create a space to give feedback and be heard
- Brainstorm solutions and improvements collaboratively
- Establish ownership of process and team practices
- Review regularly for continuous growth and modifications that fit the current needs of the team

the problem -

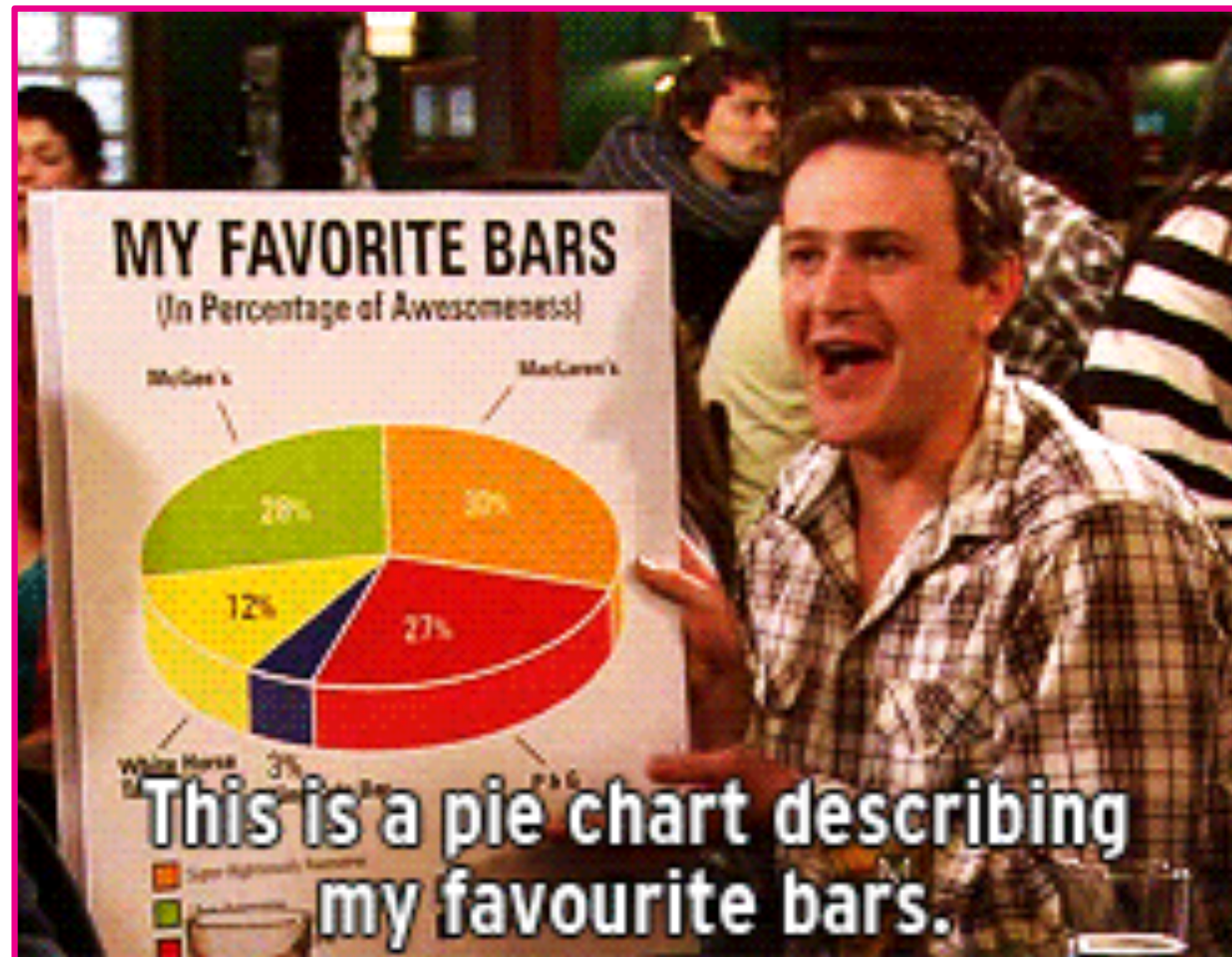


Subjective Feedback

- Personal - one sided perspective or point of view
- Individual - based on one's own experience in and outside of work
- Emotional - may include hurt, fear, anger, surprise or embarrassment
- Instinctive - knee jerk reaction triggered by fight or flight response
- Intuitive - based on insights, patterns or familiar trends

Retro Fix #1:

Quantitative Feedback



- Review the same simple metrics each retrospective
- Review trends since last retrospective
- Make targeted improvements based on data and metrics



Feelings → Data

- What impact has changes to your process had on your output? Has this gone up or down?
- How do you know that things are <fast, slow, better, worse>?
- What evidence do you have?
- What would be an indication of this problem happening again?
- How can we track if we are improving?

the problem -



Memory Bias

- Serial position effect is the tendency to recall the first and last items in a period of time better than the in the middle
- Mood congruent memory bias is the improved recall of information congruent with one's current mood
- Confirmation bias is the tendency to recall information in a way that confirms one's beliefs or hypotheses
- Recency bias is the name for the heavy influence that recent experience can have on our decision making

Retro Fix #2:

Continuous Feedback



- Gather data all the time
- Collect quant & qual
- Make it easy to collect
- Use time together to review (not remember!)



Scheduled → Continuous

- What events happened throughout the sprint?
- How could we capture events from planning to demo?
- What types of things are hardest to recall?
- What feels like the best use of our time together?

the problem -



Binary Action Items

- “Set up a meeting to discuss branch process”
- “Clean up user stories as stuff changes mid-sprint”
- “Define more clear acceptance criteria in Jira”
- “Dedicate time for developers to study topics and new tech related to their work”
- “Review sprint burnup chart as part of daily stand up”
- “Stop rotating the Scrum Master role”
- “More liberal permission to view infrastructure setting so we can resolve issues faster”
- “Include QA folks in Sprint Planning”

Retro Fix #3:

Small Experiments



- Make micro bets
 - Only one at a time
- Create a hypothesis
- Evaluate how it went before you continue



Action Item → Hypothesis

- What impact do you expect this change to have?
- How will you measure it?
- When will you check back in?
- What will you change if your hypothesis is true?
- What will you change if your hypothesis is false?



Local Optimization

- Internally focused improvements
- Things that are in our control to fix
- Improvements that directly benefit our team
- Give us control over our own process and practices
- Can create competing priorities



What is Systems Thinking?

a holistic approach that focuses on the way a system's parts interrelate and work together to contribute to the context of the larger systems.



rawpixel



Kaizen

- The Japanese translation "The act of making bad points better"
- New Oxford Dictionary definition is *"a Japanese business philosophy of continuous improvement of working practices, personal efficiency, etc."*
- The more popular translation is "change for better"
- Kaizen is a culture of learning and improving, not a practice or process



Organization Feedback

information shared with all parts of the organization to identify possible **ways to effectively work together** to achieve organizational goals.

the problem -

Single Team POV



- “The other team blocked our ability to deliver.”
- “Management makes decisions on our behalf.”
- “We are locked into dates because of the ART.”
- “We don’t have enough environments for testing.”

Scaled Retros #1:

Connect the Dots



- Share retro outputs without breaking team trust
- Review common themes and patterns
- Identify opportunities for escalation or action



Local → Global

- What data tells a more complete story about what's working and what isn't?
- Do we need people from multiple teams to come together to fix this issue?
- How can we work together to fix the big issues?
- What impact do our improvements have on other teams?
- What would be different if we all contributed to the improvement?

the problem -



Improvements in a Vacuum

- “We prioritized improving our cycle time but get blocked waiting to merge.”
- “Dependencies on other teams prevented us from addressing our sprint commitments.”
- “We can never stick to our WIP limits because of all the urgent production issues.”

Scaled Retros #2:

Create Visibility



- Create an opportunity & cadence for large scale retrospectives
- Align around key improvements
- Create visibility in action items and progress toward completion



Backlog → Transparency

- What improvements is the organization working on together?
- What changes do we expect to see for the business?
- How are individual teams contributing to moving the needle?
- How are we tracking towards our outcomes?

the problem -

Not my problem



- “They picked the <insert broken thing here>”
- “Fixing <insert broken thing here> is outside of our control”
- “We keep telling management its broken and nothing happens.”
- “We changed how we <insert broken thing here> but the other teams wouldn't so it didn't get better.”
- The dreaded...”ITS HOW WE'VE ALWAYS DONE IT.”

Scaled Retros #3:

Collective Ownership



- Understand the system impacts and the goals
- Celebrate successes as one org not just as a team
- Celebrate failures as a chance for everyone to learn and improve



Us → All of US

- What impact will this have on bringing value to our users?
- How does this align to OKRs or KPIs? Does it conflict with OKRs or KPIS?
- How could we make something that works well for us easier for another team to replicate?
- What is one improvement we can make that would help another team succeed?





Thank you!

Improving the way teams improve

colleen@scatterspoke.com