Finding and Fixing Your Organization’s Agile Potholes

Les Grove

LGrove@catalyte.io

Abstract

On the road to agility there are organizational impediments (“potholes”) that teams could run into if not removed. An agile coach could be brought in to help the organization identify and resolve those potholes. But where should a coach look to find the potholes? They could attend and observe countless ceremonies and interview dozens of team members. But it is time consuming and difficult to discover organizational impediments. A better approach is to use an agility assessment tool that easily and quickly captures data about multiple teams and guides coaching strategies.

This paper describes Catalyte’s Agile Coaching Center of Excellence process to build a simple agile assessment methodology and tool that can assess teams and their organizations. The assessment tool started as a document-based worksheet that teams incorporate as part of its retrospectives. It evolved into a spreadsheet that added scoring and measured changes over time. Then a survey form was used to better capture individual responses. The coach collects and analyzes the results from multiple teams and visually compiles the data providing pictures into each team and the organization simultaneously; helping teams at the enterprise level by identifying team and organizational impediments.

Biography

Les Grove is a Principal Agile Project Manager at Catalyte. Les has a Master’s degree in Software Engineering and holds certifications as a Certified Scrum Professional (CSP), SAFe Advance Scrum Master (SASM), and Project Management Professional (PMP).

Les has over 30 years of experience in project management, software development, and agile. He was involved with the Pacific Northwest Software Quality Conference (PNSQC) from 2002-15 as an author, reviewer, volunteer, chairperson, and board member.

This work is the result of many people who have participated in the Catalyte Agile Coaching Center of Excellence and without them the assessment tool, and this paper, would not be possible. I want to thank Eric Landes, Abby Edwards, Garrett Vonk, Adam Curtis, TJ Trujillo, Mike Gicking, Paige Hines, Rob Winkler, Torin Tierney, Temitope Olonisakin, Miguel Buddle, Danielle Schell, Sam Landesberg, Natalie Barroga, Meghan Arthur, Theo Levine, David Kaus, and David Bernal.

Introduction

Agile, from learning to adoption to practice, isn't a one-and-done exercise. It is a journey rather than a destination. With that metaphor we can equate a team with driving and navigating a car, where it needs fuel (requirements) and mechanical problems need to be discovered and repaired (inspect and adapt). Extending the metaphor further, the organization can be considered "the road" – multiple teams travel on it and any organizational impediments are the potholes that all teams end up needing to navigate around or running into. But organizations often do not see the potholes they have created for its teams. This is where an agile coach comes in. A coach can reinforce agile values and key practices while working at both the team level and organizational level.

Most agile frameworks are aimed at the team level.

- Daily standups help identify team impediments so that they can be quickly removed
- Retrospectives are vital for improving team performance and product quality
- Sprint reviews ensure the team is moving the product in the right direction

As the number of teams increase, the complexity of resolving issues at the organizational level increases exponentially because it's difficult for an organization to identify and prioritize issues at this level.

Organizational "potholes" include

- Lack of knowledge/training
- Inadequate tools
- Not using or incorrectly leveraging an agile framework
- No cross-team learning
- Non-transparency
- Poor estimation
- Over-management preventing self-direction
- Lack of business analysis capabilities
- Non-involvement of stakeholders

It becomes a situation where the loudest team or the most recent complaint gets prioritized for help from the organization.

Scaled agile frameworks have some mechanisms in place to help identify and remove organizational impediments. But these frameworks are primarily for multiple teams working on a single large product. These mechanisms don't help much for organizations that have multiple teams working for multiple products, in different domains, and for multiple clients.

Agile coaches are often brought in at the organizational level to help multiple teams. They identify organizational impediments by observing and interviewing multiple teams for several sprints and look for anti-patterns that repeatedly occur that individual teams cannot resolve on their own. This would take months.

Fortunately, there is a way for a coach to quickly gather the information they need to identify common anti-patterns: an assessment. By assessing each team, analyzing the findings and looking for opportunities that could improve multiple teams simultaneously, an organization can save time and money.

Background

The most common stumbling block to adopting and practicing agile is organizational culture and its resistance to change. To succeed in agile adoption, IT leaders need to train their organizations and reinforce agile values and key practices, create communities of practice, and deliver business outcomes.

To effectively institute cultural change, companies need to develop comprehensive hands-on agile coaching programs to assist real-world projects.

Catalyte has an Agile Coaching Center of Excellence (CoE) that provides coaching services to its teams and its clients. This group is made up of agile coaches, project managers, analysts, developers – anyone within the company who is interested in agile and coaching. The CoE was created to centralize all previous work regarding agile coaching, create coaching collateral for training and reporting, and drive initiatives to improve agile within Catalyte and its clients.

Much of Catalyte's coaching practice was based on directly observing teams, which is time-consuming and hides organizational impediments. The Agile Coaching CoE thought that a low-impact assessment would provide a quicker "lay of the land" and create opportunities for immediate recommendations.

There were several business reasons for creating a common agile assessment solution. First, Catalyte provides software development teams that work in a variety of domains: healthcare, finance, hospitality, retail, logistics, etc. Some teams are stand-alone and are managed internally; some are blended with client teams and managed by either us or the client; and others are managed by the client. The challenge is to identify teams that need coaching or expose issues that require cooperation with our clients.

Second, Catalyte provides agile coaching services for both new agile transformations and teams that are already on their agile journey. For novice organizations, assessments are like test-driven development. The initial assessment is expected to fail but improvements are expected in subsequent assessments. For mid-journey teams, assessments provide the initial snapshot of the strengths and opportunities that a coach can use to advise teams and organizations.

Lastly, Catalyte teams often need to work within a client's established agile framework. Assessments would be great to identify any organizational impediments prior to Sprint 1 and help Catalyte work with the client to mitigate risks.

Requirements for Assessments

Once the CoE decided to try assessments, the next task was to come up with some initial requirements for an assessment method.

Assessments need to be quick - People are less willing to participate in an assessment if it interferes with their sprint commitments. The longer the assessment, the less accurate the responses. We determined that an assessment shouldn't take more than 15-20 minutes to complete.

Results need to reflect the team and not individuals – Agility is about the team. Any assessment needs to focus on improving the team. This means keeping individual responses confidential in anything that is shared outside the team as well as not assessing individuals. Since the Scrum Master and Product Owner are individuals caution must be exercised when assessing the performance of those roles.

Participants need no other instructions or guidance than what's provided – Participants shouldn't have to figure out what's expected of them when completing the assessment. Observational trials can help, but always expect opportunities for assessment improvements based on feedback.

Results identify opportunities for team improvement – A team will be more likely to participate if they get something for their efforts. The team results can be something they can use towards their own excellence.

Aggregate results to identify organizational impediments – Aggregated results can show patterns that may indicate tools, process, or training issues that affect many or all teams. Teams may not even be aware as they are more focused internally.

Participants need to be able to provide details on their responses – To make assessments quick, simple responses are required, but at the cost of understanding why best practices aren't used. A more detailed feedback mechanism needs to be in place during or after the assessment. This, in turn, feeds back into the assessment.

Identify improvements or regressions over time – Follow-up assessments provide additional snapshots to see whether teams are improving, staying stagnate, or getting worse.

Accommodate any flavor of agile, not just Scrum – Even though most teams practice Scrum, it’s not a guarantee. An assessment needs to be applicable to many flavors of agile, including Kanban.

Building the Assessment Tool

In building an assessment tool, we wanted to focus on getting the tool right before covering all agile topics. We decided on five initial assessment topics to be covered in Version 1 of the tool. We felt that these were a good initial sampling of topics covering issues that we have seen in our projects at Catalyte. The five topics were:

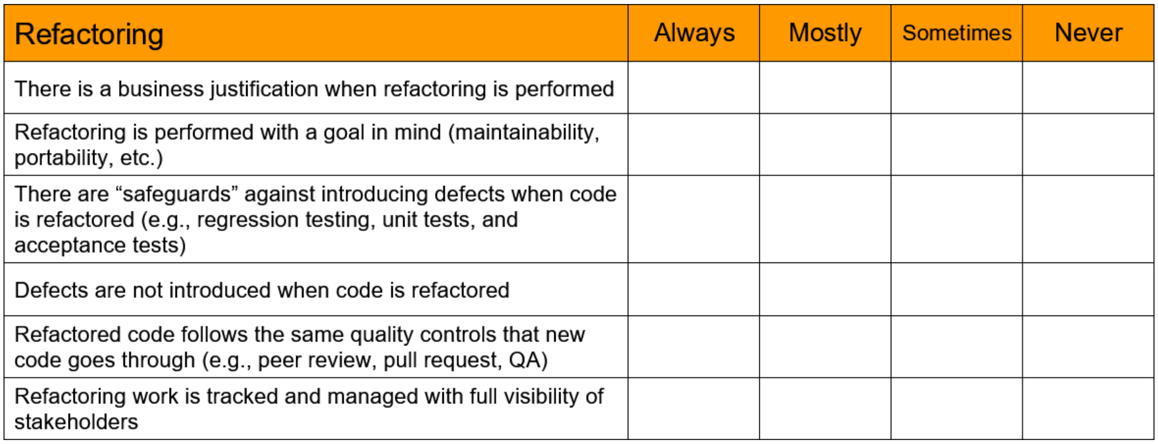
- Team Integrity (ability to get its work done without external interference)
- Retrospectives
- Requirements
- Backlog Management
- Refactoring

Next, we determined a method for the assessment. We chose to have five-to-seven statements for each topic, where each statement represented an intended practice or outcome. To make the assessment easy to take and to generate comparative results, we chose a Likert scale: “Always”, “Mostly”, “Sometimes”, and “Never” where “Always” would be the most positive response.

For each topic, most of five-to-seven statements were written from scratch then reviewed and revised in the CoE several times. Others were pulled from other existing assessments [1] and revised to fit our goals and format.

1.1 Version 1 – Word Document

For the first iteration, we were more interested in piloting the topics, method, and statements than making the tool a great user experience. So, for executing the assessment we created a simple Word document that teams would use during one of their retrospectives. The document included instructions and places for feedback on the assessment itself.

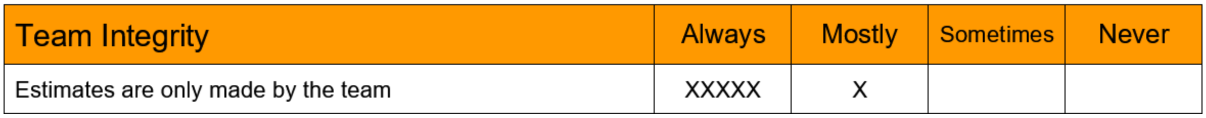


Refactoring	Always	Mostly	Sometimes	Never
There is a business justification when refactoring is performed				
Refactoring is performed with a goal in mind (maintainability, portability, etc.)				
There are “safeguards” against introducing defects when code is refactored (e.g., regression testing, unit tests, and acceptance tests)				
Defects are not introduced when code is refactored				
Refactored code follows the same quality controls that new code goes through (e.g., peer review, pull request, QA)				
Refactoring work is tracked and managed with full visibility of stakeholders				

Figure 1 Early survey items for Refactoring topic

Figure 1 shows the survey section for the Refactoring topic. Teams would fill it out together, usually during the Gather Data part of their retrospective [2], giving them an opportunity to discuss areas where they were not responding with “Always” or “Mostly”. This version of the assessment tool was piloted with six teams.

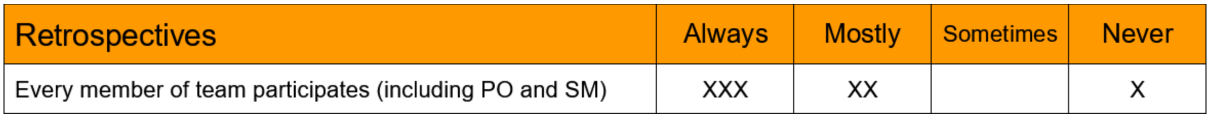
From an organizational perspective, Figure 2 shows an agile practice that is usually being done across the organization.



Team Integrity	Always	Mostly	Sometimes	Never
Estimates are only made by the team	XXXXX	X		

Figure 2 Successful Agile Results

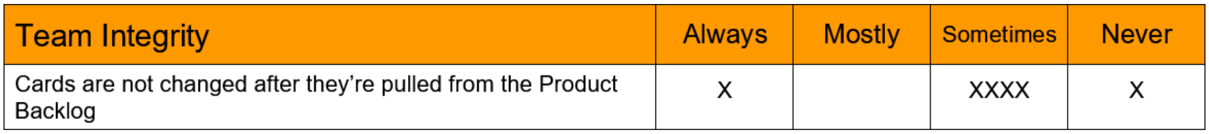
From a team coaching opportunity perspective, Figure 3 shows that one team is not having full participation in retrospectives. A team coach or scrum master can investigate where this team is taking any actions to help improve this.



Retrospectives	Always	Mostly	Sometimes	Never
Every member of team participates (including PO and SM)	XXX	XX		X

Figure 3 Single Team Opportunity Identified

When looking at Figure 4, only one team stated that they were freezing story cards, but the other five teams were indicating that in-sprint story cards are often being changed during sprints. A coach working at the organizational level, would want to dig into this a little deeper to see if there are any organizational impediments that teams are working against. In this example, one team stated that they were always preventing card changes, but it might be worth the effort to check with that team to see if that’s truly the case – they could have misunderstood the statement or found a way to overcome the organizational impediment that other teams could leverage.



Team Integrity	Always	Mostly	Sometimes	Never
Cards are not changed after they’re pulled from the Product Backlog	X		XXXX	X

Figure 4 Organizational Pothole Discovered?

There were two main drawbacks to this Word-based version of the tool. First, it was time-consuming to compile the results by hand – and therefore error-prone and not very scalable. Second, there was a challenge to show team and organizational improvements or regressions after repeating the assessments.

1.2 Version 2 – Spreadsheet

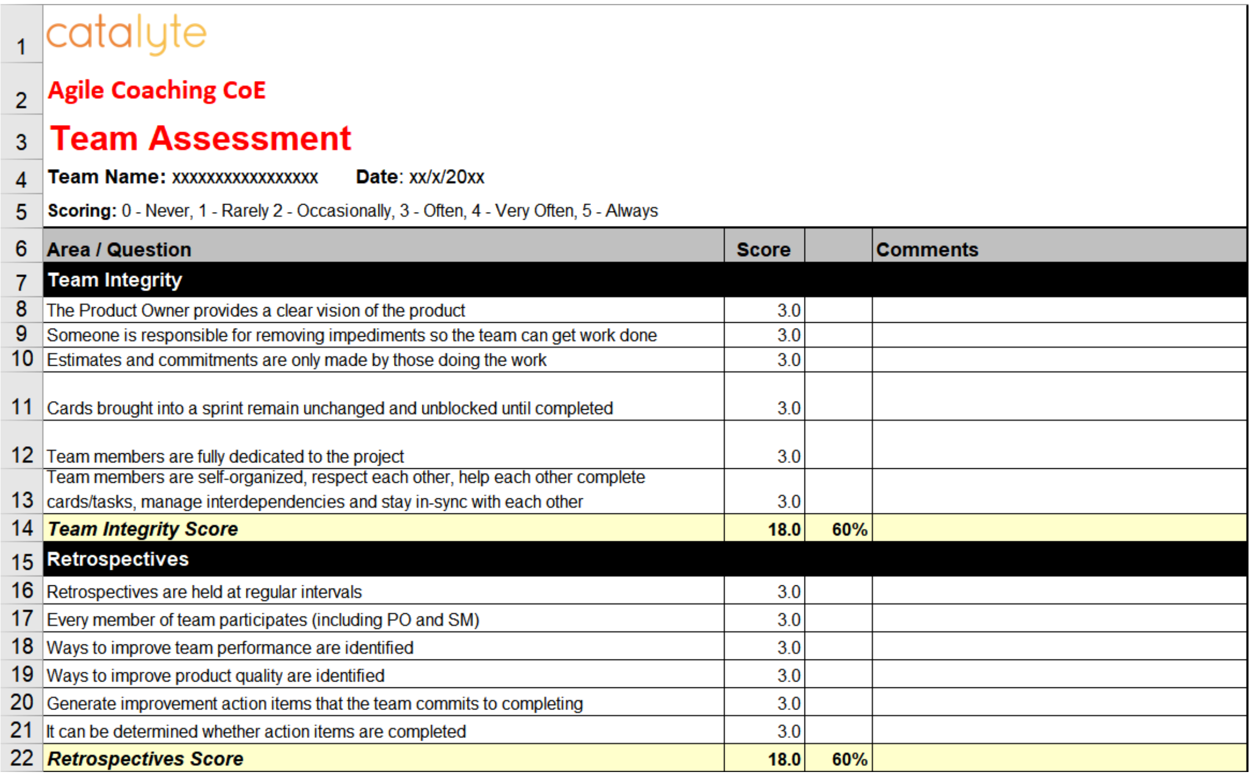
One of the CoE members took a Scaled Agile Framework (SAFe) course and discovered the SAFe Team Self-Assessment [5]. The similarities surprised us as both our home-grown assessment and the SAFe Team Self-Assessment covered five topics and had a similar number of statements per topic.

The CoE also liked these additional features:

- More refined choices of responses that allowed for more precise measurement
- A scored response for each statement
- A combined percentage score for the topic providing indications of change in subsequent assessments

- A comment section for each statement that provided details at the statement response level rather than at a topic level

The CoE decided not to use the SAFe spreadsheet statements because it covered some topics, like PI Health, not relevant in non-scaled situations. Also, its terminology was very SAFe-specific and something not necessarily familiar to most participants. However, some statements from the SAFe assessment form were added to Version 2 of Catalyte's assessment (see Figure 5).



1 catalyte

2 Agile Coaching CoE

3 Team Assessment

4 **Team Name:** xxxxxxxxxxxxxxxxx **Date**: xx/x/20xx

5 **Scoring:** 0 - Never, 1 - Rarely 2 - Occasionally, 3 - Often, 4 - Very Often, 5 - Always

6	**Area / Question**	**Score**		**Comments**
7	**Team Integrity**			
8	The Product Owner provides a clear vision of the product	3.0		
9	Someone is responsible for removing impediments so the team can get work done	3.0		
10	Estimates and commitments are only made by those doing the work	3.0		
11	Cards brought into a sprint remain unchanged and unblocked until completed	3.0		
12	Team members are fully dedicated to the project	3.0		
13	Team members are self-organized, respect each other, help each other complete cards/tasks, manage interdependencies and stay in-sync with each other	3.0		
14	***Team Integrity Score***	**18.0**	**60%**	
15	**Retrospectives**			
16	Retrospectives are held at regular intervals	3.0		
17	Every member of team participates (including PO and SM)	3.0		
18	Ways to improve team performance are identified	3.0		
19	Ways to improve product quality are identified	3.0		
20	Generate improvement action items that the team commits to completing	3.0		
21	It can be determined whether action items are completed	3.0		
22	***Retrospectives Score***	**18.0**	**60%**	

Figure 5 Assessment Spreadsheet Derived from SAFe

Unlike Version 1's table format, Version 2's spider chart (see Figure 6) now shows all team scores in all topics in one place.

There are three key learnings from the chart in Figure 6:

- Team 3 did well with requirements. Need to find out what the other teams can learn
- Team 2 needed to improve their retrospectives.
- Teams 5 and 6 scored relatively low overall

As an agile coach, there is plenty of data to help individual teams and to find some areas to improve at the organizational level:

- Why were Requirements so inconsistent?
- Why weren't there any teams that scored at the highest levels for Team Integrity and Refactoring?
- What was going wrong with Teams 5 and 6?
- Why were half the teams underperforming in Retrospectives?

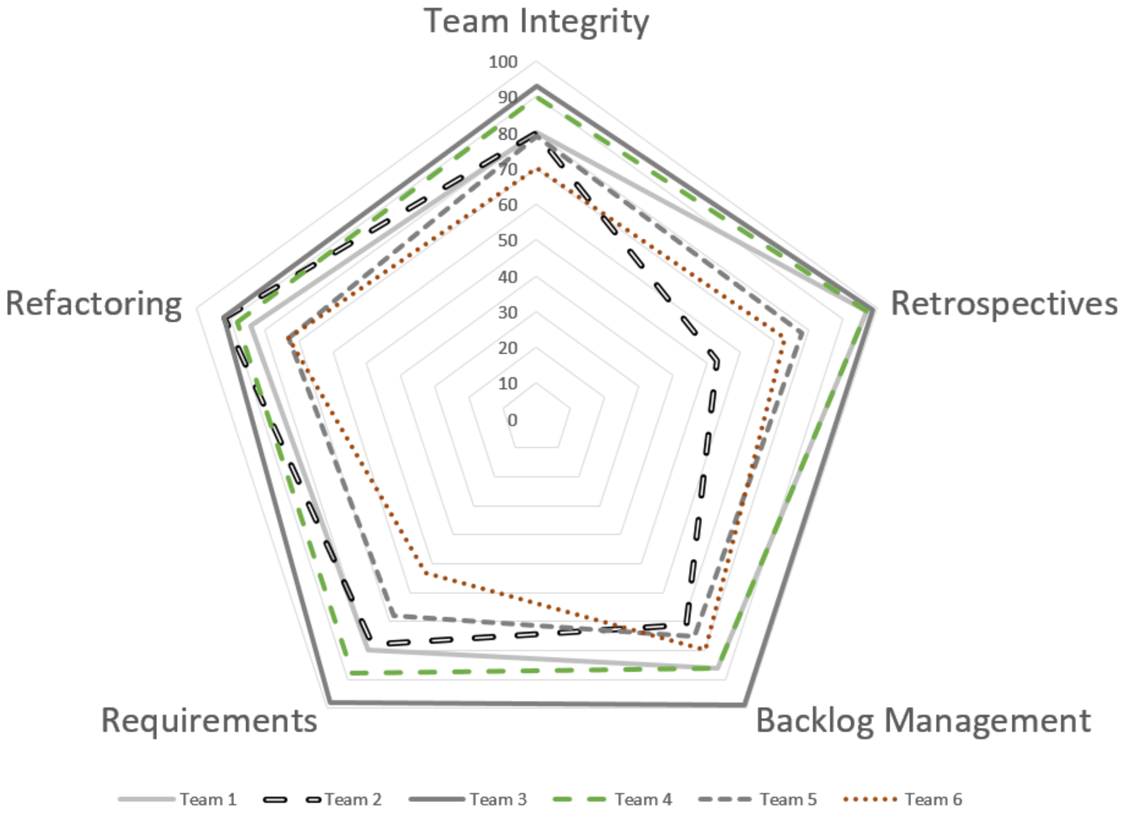


Figure 6 Spider Chart for Version 2 Assessment

1.3 Version 3 – Google Form

One of the teams took the Version 2 spreadsheet and created a Google Form and had each team member fill out the survey individually then they met as a team, reviewed the results, and formed a consensus response on the spreadsheet. The CoE liked that concept and used the form for the next round of assessments. The thinking was that responding to surveys might minimize the effect of strong, dominating voices pushing the responses higher in a group setting.

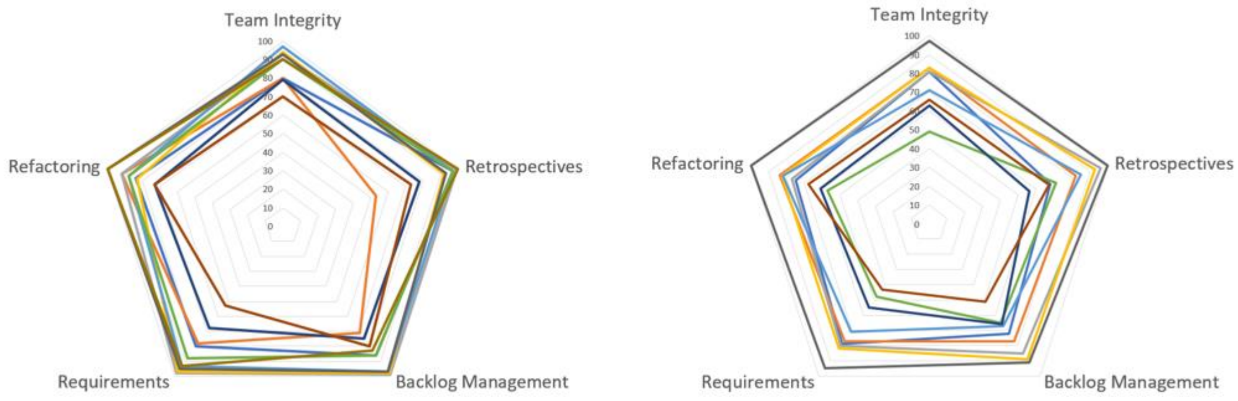


Figure 7 Comparison of single team response vs averaging individual responses

We found that when we compare the results using this method with the previous results the data points were generally lower and more spread out (see Figure 7). There are two key advantages to surveying individuals. First, it's easier to identify the organizational potholes. For example, Backlog Management looks to be an area that is inconsistently performed. Second, a large spread of responses within a team to the same statement could be a signal a lack of common understanding, a practice that is not consistently done within the team, or an ambiguously written assessment statement.

1.4 Version 4 – Expanded Topics

For the next iteration of the assessment tool, our intent was to increase the number of topics covered. We went to some of Catalyte's other CoEs (Architecture, DevOps, and Quality) and asked them for a set of assessment statements to include in the tool. Any results from these topics would be shared directly with the respective CoE for their own analysis and follow-up.

For the Quality topic, the following statements were included in the survey:

- Our test coverage satisfies our test quality goals
- Critical defects are prevented from reaching Production
- Our product is stable
- The tests are automated where possible
- Test results are used as the basis of code integration and deployment

Figure 8 shows an analysis of teams that have potential coaching needs with respect to quality.

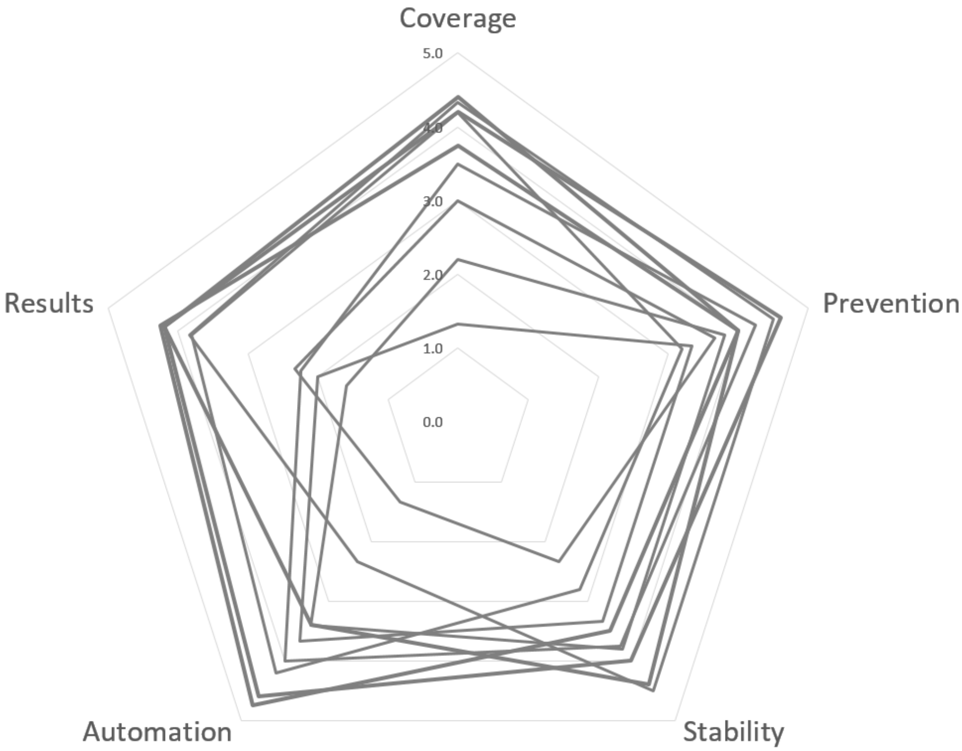


Figure 8 Analysis of Quality Topic (Team Subset)

There are some key coaching opportunities when look at the survey results:

- There are some teams doing well with test automation but there are some other teams that are doing poorly. This would be a great time for knowledge sharing between the strong teams and the struggling teams.

- In looking for organizational improvements, there seems to be some barriers preventing any teams from scoring well in test code coverage and using test results in integration and deployment.

Next Steps

One of the challenges of an internally developed, custom tool is that a lot of extra effort is required to administer assessments, analyze the data, and share results with the teams. This is fine for a handful of teams, but this will not scale and may not be as professional when we assess client teams. The Agile Coaching CoE is piloting the Comparative Agility tool (comparativeagility.com). Although it has its own topics and statements, and it doesn't meet some of the original requirements, the CoE feels it is worth it to see if the added value is greater than the requirements not met.

Valuable features include:

- Easily compare team results with its previous results, with the organization, or with a world index of all teams that have taken the assessment
- Quick and easy reports that are also interactive
- Visual representation of the distribution characteristics for each statement to show where there are large disagreements within the team

Comparative Agility covers the quality topic with these statements:

- Product owners actively participate in the creation of the acceptance criteria for each feature.
- All bugs are fixed during the iteration in which they are found.
- At the end of each iteration there is little or no manual testing required.
- The team performs a variety of types of testing including functional, performance, integration, and scalability each iteration.
- Team members who perform testing are involved and productive right from the start of each iteration.
- At the end of each iteration, the team has high-quality working software that it is comfortable being tested by people outside of the team.
- The team has pre-defined and agreed-upon criteria for considering a feature done.

Summary

Our journey in building our Agile Assessment Tool, just like our teams' journeys to agility, it's never ending. Catalyte's Agile Coaching Center of Excellence has learned as much about assessments as has been learned about the teams. We've learned

- The faster and more accurate we can process assessment data, the better we, as coaches, can help teams and organizations
- Visual representation of data helps identify coaching opportunities
- Multiple individual survey responses are often better than single team responses in identifying areas for deeper investigation

With better visibility of the organizational road we can find and fix the organizational potholes and help teams go faster in their journey to agility.

References

1. Ben Linders Consulting, “How Agile Are You?” https://www.benlinders.com/tools/agile-self-assessments/ (accessed July 6, 2018).

2. Derby, Ester, Larsen, Diana, 2006 *Agile Retrospectives: Making Good Teams Great*, Pragmatic Bookshelf

3. Gartner Research, Market Guide for Agile and DevOps Services, Oct 2017

4. Grove, Les. 2002. “Charting the Course through Software Process Improvement.” PNSQC Proceedings (2002).

5. Scaled Agile Framework, “SAFe Team Self-Assessment”, https://www.scaledagileframework.com/metrics/#T4 (accessed July 20, 2018)